

## Final Report

# PROGRAMMABLE DATA COMMUNICATIONS CONTROLLER REQUIREMENTS

(NASA-CR-150384) PROGRAMMABLE DATA N77-31859  
COMMUNICATIONS CONTROLLER REQUIREMENTS  
Final Report (Teledyne Brown Engineering)  
61 p HC A04/MF A01 CSCI 09B Unclass  
G3/62 47390

July 1977

REPRODUCED BY  
NATIONAL TECHNICAL  
INFORMATION SERVICE  
U. S. DEPARTMENT OF COMMERCE  
SPRINGFIELD, VA. 22161

 **TELEDYNE  
BROWN ENGINEERING**

Cummings Research Park • Huntsville, Alabama 35807

FINAL REPORT  
SD77-MSFC-2115

PROGRAMMABLE DATA COMMUNICATIONS  
CONTROLLER REQUIREMENTS

July 1977

Prepared For

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
GEORGE C. MARSHALL SPACE FLIGHT CENTER  
HUNTSVILLE, ALABAMA

Contract No. NAS8-31488

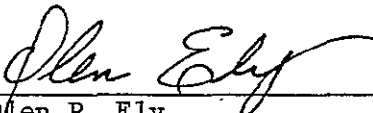
Prepared By

SYSTEMS DIVISION  
TELEDYNE BROWN ENGINEERING  
HUNTSVILLE, ALABAMA

## ABSTRACT

This report presents the design requirements for a Programmable Data Communications Controller (PDCC) that reduces the difficulties in attaching data terminal equipment to a computer. The PDCC is an interface between the computer I/O channel and the bit serial communication lines. Each communication line is supported by a communication port (CP) that handles all line control functions and performs most terminal control functions. The port is fabricated on a printed circuit board that plugs into a card chassis, mating with a connector that is joined to all other card stations by a data bus. Ports are individually programmable; each includes a microprocessor, a Programmable Read-Only Memory (PROM) for instruction storage, and a Random Access Memory (RAM) for data storage.

APPROVED:

  
\_\_\_\_\_  
Glen P. Ely  
Project Director

## TABLE OF CONTENTS

		Page
1.	INTRODUCTION . . . . .	1-1
2.	RATIONALE FOR THE PDCC CONCEPT . . . . .	2-1
	2.1 A Typical Data Communication Subsystem . . . . .	2-1
	2.2 The Programmable Data Communication Controller . .	2-4
3.	THE PDCC CONFIGURATION . . . . .	3-1
	3.1 The Crate and Plug-In Units . . . . .	3-3
	3.2 Wiring of PDCC Station Connectors . . . . .	3-3
	3.3 Control Unit (CU) . . . . .	3-5
	3.4 Communication Port (CP) . . . . .	3-17
4.	OPERATION OF THE PDCC . . . . .	4-1
	4.1 Interchange of Control and Status Information . .	4-1
	4.2 Block Transfer of Communication Data . . . . .	4-9
	4.3 Functional Description of a Communication Port . .	4-14
5.	RECOMMENDED INVESTIGATIONS . . . . .	5-1

# LIST OF ILLUSTRATIONS

Figure	Title	Page
2-1a	Computer Terminal Configuration Meeting Current Needs . . . . .	2-2
2-1b	Computer Terminal Configuration Meeting Future Needs . . . . .	2-2
2-2	Typical Multi-Terminal System Configuration . . . . .	2-3
2-3	Hardware/Software Telecommunications Configuration using PDCC . . . . .	2-6
3-1	PDCC Configuration . . . . .	3-2
3-2	PDCC Wiring Diagram . . . . .	3-4
3-3	Contact Assignments for CP Stations . . . . .	3-6
3-4	Contact Assignments for CU (Station 24) . . . . .	3-7
3-5	Control Unit (CU) Schematic Block Diagram . . . . .	3-9
3-6	CU Microcomputer . . . . .	3-10
3-7	Station Select Unit . . . . .	3-11
3-8	CU Microcomputer I/O . . . . .	3-13
3-9	Look-At-Me Register . . . . .	3-14
3-10	Station Control Register (SCR) and Station Status Register (SSR) for CU . . . . .	3-16
3-11	Plug-In Board at Station N . . . . .	3-18
3-12	Gating Logic at Each Station . . . . .	3-20
3-13	Station Control Register (SCR) and Station Status Register (SSR) . . . . .	3-21
3-14	CP Microcomputer . . . . .	3-24
3-15	CP Microcomputer I/O . . . . .	3-25
3-16	Input Command Register (ICR) . . . . .	3-26

# LIST OF ILLUSTRATIONS - Concluded

Figure	Title	Page
3-17	Output Command Register (OCR) . . . . .	3-27
3-18	μC Control Register and μC Status Register in CP . . . .	3-29
3-19	CP Shared Memory . . . . .	3-31
4-1	Functional Diagram of Command Interchange . . . . .	4-2
4-2	Functional Diagram of Block Transfer Operation . . . .	4-10
4-3	Data Flow Internal to Communication Port . . . . .	4-15

# 1. INTRODUCTION

The National Aeronautics and Space Administration (NASA) uses many computers, large and small, at its space flight centers. Increasingly, data terminal equipment is being connected to these computers to significantly increase the effectiveness of their utilization. Adding terminals can be frustrating and expensive, however. The NASA inventory includes the products of many manufacturers. Unfortunately, data communication subsystems are developed by each computer manufacturer for its line of computers, and as a further complication, these communication products are likely to be specialized for a specific line protocol (e.g., Bi-Sync, SDLC, ASCII Asynchronous). Consequently, connecting a new terminal to a specific computer may require a significant investment in hardware and software.

This report presents the design requirements for a Programmable Data Communications Controller (PDCC) that reduces the difficulties in attaching data terminal equipment to a computer. The PDCC is an interface between the computer I/O channel and the bit serial communication lines. Each communication line is supported by a communication port (CP) that handles all line control functions and performs most terminal control functions. The port is fabricated on a printed circuit board that plugs into a card chassis, mating with a connector that is joined to all other card stations by a data bus. Ports are individually programmable; each includes a microprocessor, a Programmable Read-Only Memory (PROM) for instruction storage, and a RANDOM Access Memory (RAM) for data storage.

Three facts make the PDCC concept an attractive and feasible approach for connecting data terminal equipment to a computer. First, the ports are independent physical units that can be easily added or removed without modifying any other components of the PDCC; consequently, they can be used in any PDCC without regard to what computer is the host. Second, the PDCC's distributed processor architecture simplifies the host computer's control functions to the point that special line and terminal

control programs are unnecessary. Third, the trend in the large-scale integration (LSI) of electronic circuits is toward a very low cost for a small, simple computer -- so low, in fact, that programmable communication ports will be economically practical.

Subsequent sections of this report discuss the rationale for the PDCC in more detail (Section 2) and describe the design requirements and performance (Sections 2 through 4). The reader will benefit from having two companion documents available for immediate reference. They are:

1. "IEEE Standard Modular Instrumentation and Digital Interface System (CAMAC)", IEEE Std 583-1975, Institute of Electrical and Electronic Engineers, 1975
2. "Intel 8080 Microcomputer Systems User's Manual", Intel Corporation, 1975.



## 2. RATIONALE FOR THE PDCC CONCEPT

The PDCC concept solves several problems that cause difficulties in connecting data terminal equipment to a computer. It reduces the amount of special-purpose interfacing hardware and the amount of special-purpose I/O software required to configure a working system. Further, the PDCC concept enhances the system designer's flexibility in interconnecting multi-computer/multi-terminal systems. It also shortens the time from system concept to working system.

Adding a new terminal can be a major investment if its line protocol is unlike other terminals in the system. Further, the problem reappears each time the same terminal is added to a different type computer. Figure 2-1 depicts the situation that can arise in a facility where several computers of different manufacture are providing services to a number of terminals (Figure 2-1a). Having the flexibility to reconfigure this arrangement (Figure 2-1b) means that each computer must have all the data communication equipment and software necessary for connecting every type of terminal. None of the equipment is interchangeable because each computer manufacturer produces multiplexers and line adapters that are applicable only to its product line. In the extreme, a total of  $n \times m$  controllers is needed to make arbitrary connection of  $m$  distinct terminals to  $n$  different computers.

This section examines the deficiencies of current data communication subsystems and introduces a different approach: the Programmable Data Communications Controller.

### 2.1 A TYPICAL DATA COMMUNICATION SUBSYSTEM

Figure 2-2 shows a typical configuration of equipment for a multi-terminal environment. The units labeled "controller" in the diagram may be either a multiplexer and associated line adapters or individual line controllers. In either case, much of the equipment is specialized for a particular line protocol.

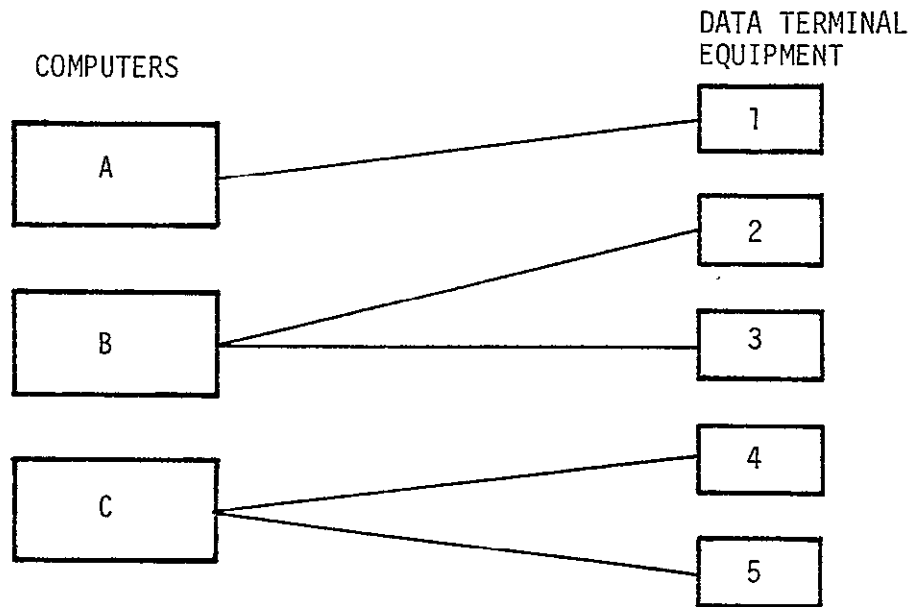


FIGURE 2-1a. COMPUTER TERMINAL CONFIGURATION MEETING CURRENT NEEDS

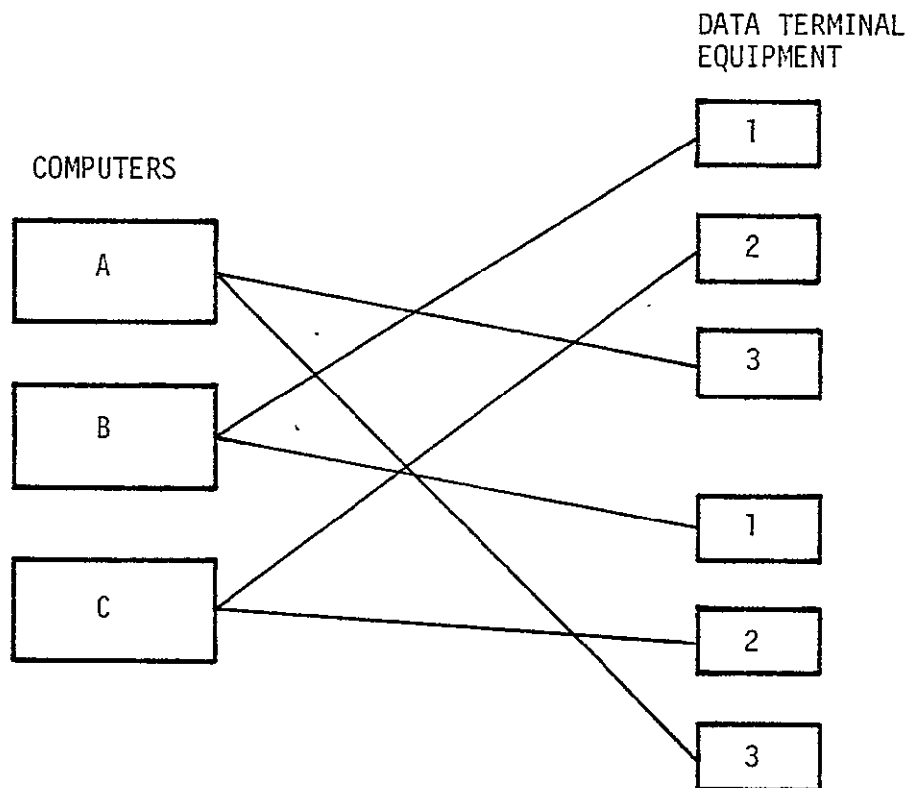


FIGURE 2-1b. COMPUTER TERMINAL CONFIGURATION MEETING FUTURE NEEDS

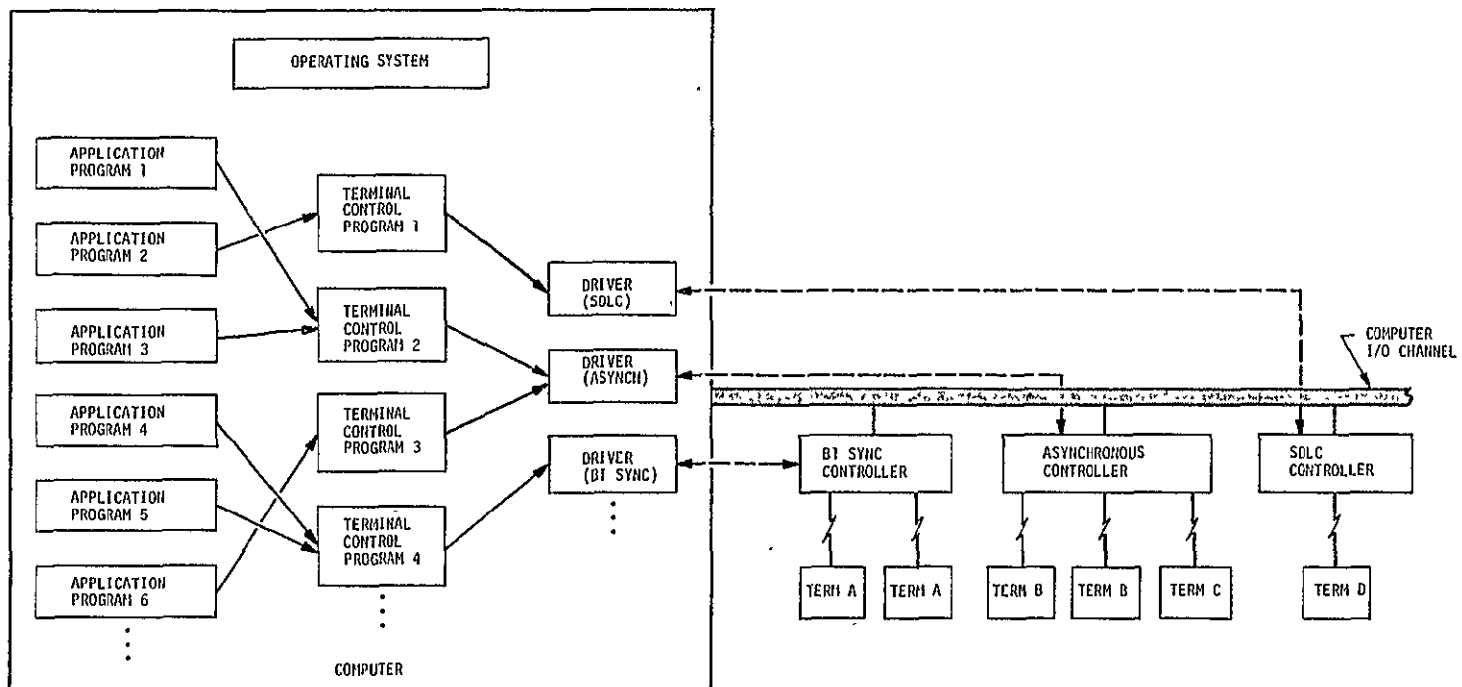


FIGURE 2-2. TYPICAL MULTI-TERMINAL SYSTEM CONFIGURATION

The computer operating system includes a driver for each different type of controller, as shown in Figure 2-2. Further, there is a terminal control program for each distinct type of terminal. As an example, assume that the terminals shown on the asynchronous controller include one display terminal that emulates a teletypewriter (a "dumb" terminal) and another that is fully buffered. Because both use the ASCII asynchronous line protocol, a single driver handles the input/output of characters to the two terminals. On the other hand, the terminal control characteristics of the two are quite different. The dumb terminal transmits every key stroke including backspaces, etc., whereas the buffered terminal transmits blocks of data after editing operations are complete.

If the application programs that use either terminal communicate directly with the driver, they must contain all the terminal control protocol for one or both of the display terminals. The alternative, as shown in Figure 2-2, is an operating system that includes two terminal control programs: one performs terminal-specific functions for the dumb terminal (e.g., echo of characters, suppression of characters that are cancelled by a backspace) and the other performs terminal-specific functions for the buffered terminals (e.g., generating and removing field delineators, unblocking page images).

The principal disadvantage of the configuration shown in Figure 2-2 is its specialization to a particular computer and to a specific line protocol. The line control functions are distributed between a controller and its software driver, and the terminal control functions are distributed between the driver, a terminal control program, and the application program. The PDCC, on the other hand, is based on the principle that it performs all line control functions and most terminal control functions. Consequently, once the PDCC is interfaced to a computer, all the terminals supported by the PDCC can be connected.

## 2.2 THE PROGRAMMABLE DATA COMMUNICATION CONTROLLER

An obvious way to simplify the hardware/software configuration of Figure 2-2 is to draw the diagram using a single apparatus that performs

the functions of all the communication controllers. Figure 2-3 is a revised diagram with the PDCC controller. The major advantage of this configuration is the potential for reducing the number of interfaces in a network of computers and terminals. Interfacing the PDCC to a computer also interfaces all the terminals that the PDCC has been programmed to handle. The maximum number of computer/terminal interfaces is  $m + n$  with the PDCC (rather than the  $m \times n$  for conventional communication subsystems).

Of course, having a programmable controller does not assure flexibility; the PDCC must have a high degree of modularity. The required flexibility can be achieved, however, by controlling each communication line with a separate device that can be easily added or replaced to accommodate additional or alternative terminals. Recent advances in integrated circuits -- particularly the advent of microprocessors, solid-state memories, and related devices -- created the possibility that the needed modularity can be achieved economically. The investigation has emphasized the use of microprocessors and the microprocessor chip families in the belief that current and/or future costs will favor the distribution of computing functions to achieve flexibility.

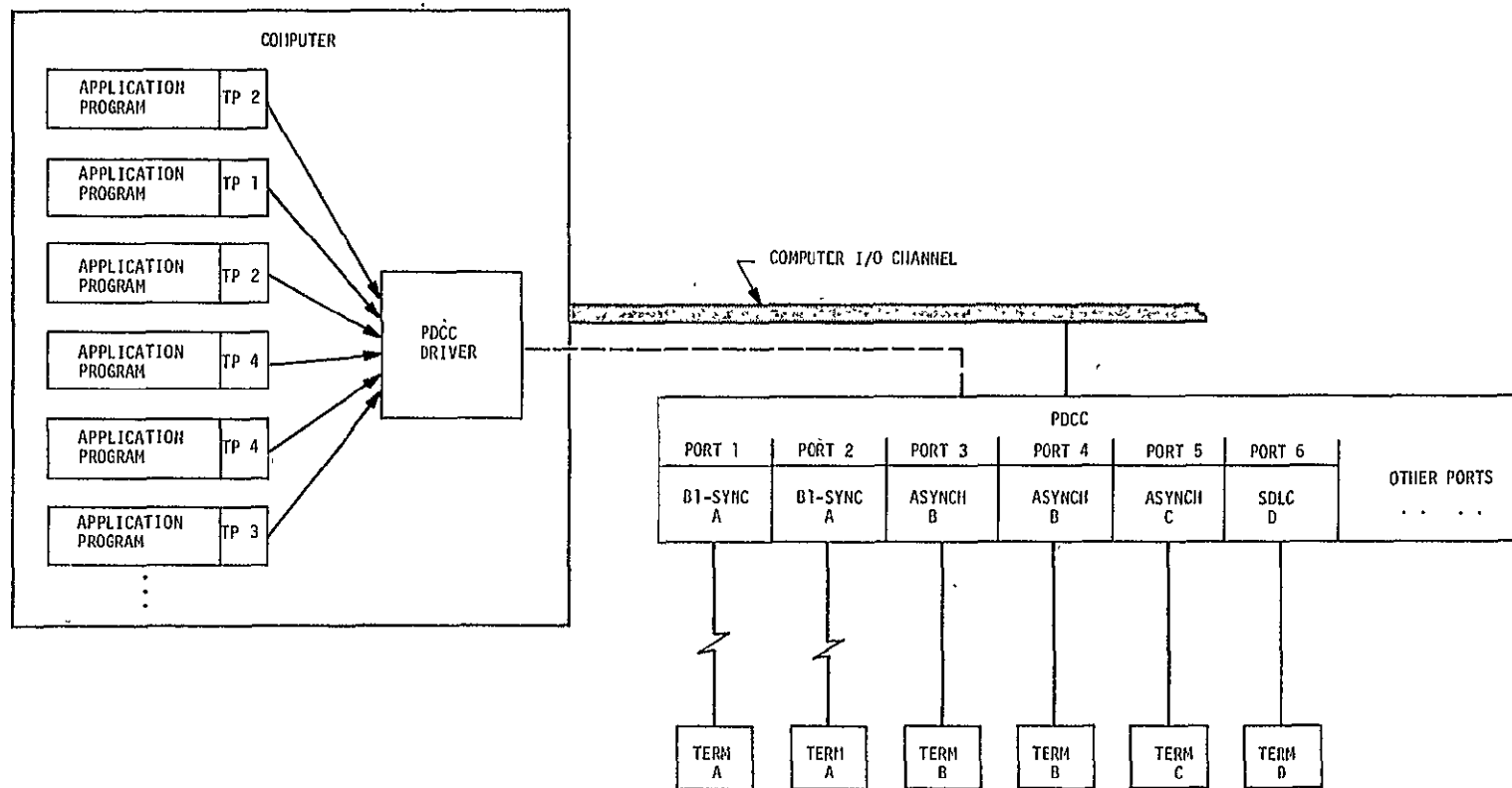


FIGURE 2-3. HARDWARE/SOFTWARE TELECOMMUNICATIONS CONFIGURATION USING PDCC

### 3. THE PDCC CONFIGURATION

The basis for establishing the design requirements of the PDCC is the IEEE Standard Modular Instrumentation and Digital Interface System (CAMAC). The equipment assembly is formed by mounting plug-in units in the CAMAC crate. Each communication port (CP) is a circuit on one of the plug-in units that occupies a mounting station in the crate. A control unit (CU) regulates operation of the PDCC and contains an interface to the host computer's I/O channel. It is fabricated on a double-width circuit board that occupies two adjacent mounting stations. Figure 3-1 is a schematic block diagram of the PDCC, showing:

- The control unit
- Up to 23 CP stations
- A common power supply
- Station connectors
- The Databus
- Communication connectors for each port.

At each station is a connector socket with 86 contacts. At normal stations (Stations 1 through 23), the connector gives access to the PDCC Databus, which is an assembly of control, data, and power lines that connect to the same contact on the sockets at all normal stations. The Databus utilizes 36 contacts on each station connector. The remaining 48 contacts are available for attaching cables to standard communication connectors (RS 232C, RS 423, EIA current loop, etc.), which are also mounted in the crate. Two contacts are unused.

Each communication port is programmable. It transmits messages to a terminal and receives messages from the terminal via bit-serial communication lines. The control unit communicates selectively with each CP using standard single-word messages. It performs the control and multiplexing functions to transport communication messages between the host computer and the ports. The messages are transferred incrementally in data blocks of 64 bytes.

Subsequent sections describe the mechanical and electrical characteristics of the PDCC. The presentation presumes a familiarity with the CAMAC Standard, and the reader can benefit by having that publication

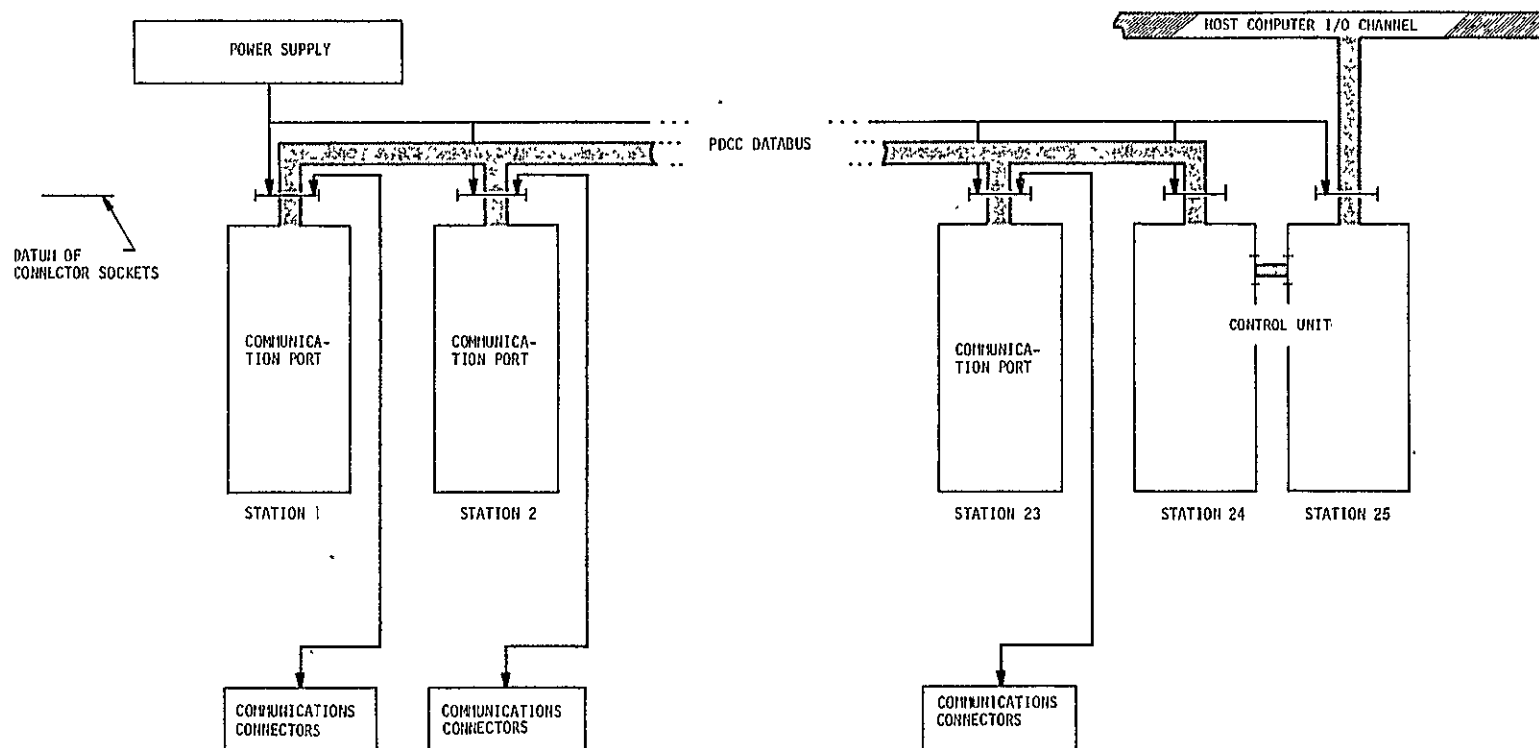


FIGURE 3-1. PDCC CONFIGURATION



handy. Although the functional characteristics of the PDCC Databus are quite different from the CAMAC's Dataway, most of the mechanical layout and some of the wiring is very similar.

### 3.1 THE CRATE AND PLUG-IN UNITS

The mechanical characteristics of the PDCC crate are identical to CAMAC except for the additional requirements related to the communication connectors. Consequently, certain sections of the CAMAC Standard apply directly to the PDCC. In particular, Section 4.1 ("The Crate"), Section 4.2 ("Plug-In Units"), and supporting figures, 1 through 6 and Figure 9, are included as design requirements for the PDCC. The physical dimensions of the crate and the printed wiring cards, the specifications for the connectors, and the clearances for free access are identical.

The CAMAC Standard requires that each plug-in unit have a front panel with a fixing screw. There seems to be no necessity for this requirement if the crate is mounted internal to another enclosure (e.g., a computer cabinet). Consequently, the use of front panels on the plug-in units is optional.

The CAMAC Standard provides for an Interface Housing Unit for extending the depth of a CAMAC crate (Figures D2 and D5, CAMAC Standard). This unit is required for the PDCC. It is needed to provide space for the communication connectors and cables associated with each Communication Port. The communication connectors shown in Figure 3-1 can be mounted in the Interface Housing Unit or integrated into a connector plane using techniques employed in some computer equipment (e.g., the Varian 77-350X cardframe chassis for the V-77 series computers).

### 3.2 WIRING OF PDCC STATION CONNECTORS

Figure 3-2 shows the wiring of connector sockets in a PDCC that has a full complement of 25 stations. The 23 stations occupied by the communication ports (i.e., Stations 1 through 23) are wired identically as follows:

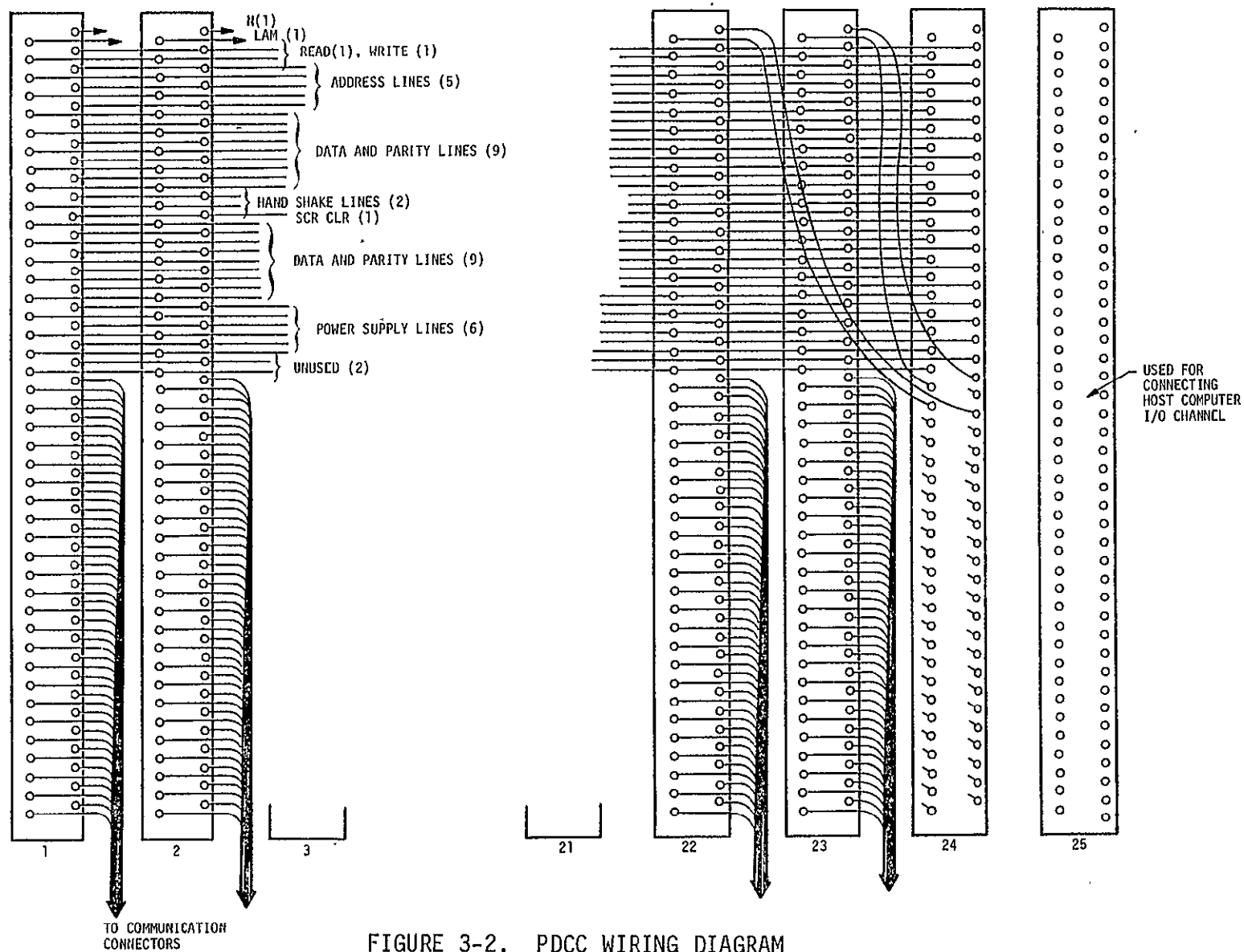


FIGURE 3-2. PDCC WIRING DIAGRAM

- Two individual lines connecting the CP station to the connector at Station 24
- 34 bus lines that link corresponding contacts at these stations and at Station 24
- 48 lines to separate communication connectors.

The wiring of Stations 24 and 25 (the control unit) is different from that of CP stations. As noted above, the bus lines that connect to the CP stations also connect to Station 24 but not Station 25. Further, 46 of the contacts on the Station 24 connector socket are used to attach the pairs of individual lines to each CP station. All contacts at Station 25 are reserved for connecting the host computer's I/O channel.

The 36 bus lines and the 46 individual lines form the PDCC Databus, which is the highway for transporting data, commands, and control signals between the control unit and the communication ports.

The table in Figure 3-3 shows the contact allocation for the 23 CP stations, and the table in Figure 3-4 shows the contact allocation for Station 24 of the control unit. The assignments are given in terms of five wiring groups:

- Command lines
- Block transfer lines
- Power distribution bus
- Individual lines
- Interface lines (for use with the communication connectors).

Section 4 explains how the Databus is used in operation of the PDCC.

### 3.3 CONTROL UNIT (CU)

The control unit occupies the two right-most mounting stations in the crate (i.e., Stations 24 and 25). It consists of the following:

## COMMAND LINES



### BLOCK TRANSFER LINES



## POWER SUPPLY LINES



## INTERFACE LINES



STATION SELECT

```
WRITE
RESERVED (SUBSTATION ADDRESS)
REGISTER ADDRESS
REGISTER ADDRESS
DATA
DATA
DATA
DATA
```

```

HANDSHAKE
CLEAR SCRs
DATA
DATA
DATA
DATA

```

+5 dc  
0 dc (POWER RETURN)  
+12 dc

UNUSED

## CONNECTOR

[illegible]

LOOK-AT-ME

```

READ
RESERVED (SUBSTATION ADDRESS)
REGISTER ADDRESS
RESERVED (PARITY)
DATA
DATA
DATA
DATA

```

```
HANDSHAKE
RESERVED (PARITY)
DATA
DATA
DATA
DATA
```

-5 dc  
0 dc  
-12 dc

UNUSCD

INDIVIDUAL LINE

## COMMAND LINES



### BLOCK TRANSFER LINES



## POWER SUPPLY LINES



## INTERFACE LINES



FIGURE 3-3. CONTACT ASSIGNMENTS FOR CP STATIONS

# CONNECTOR

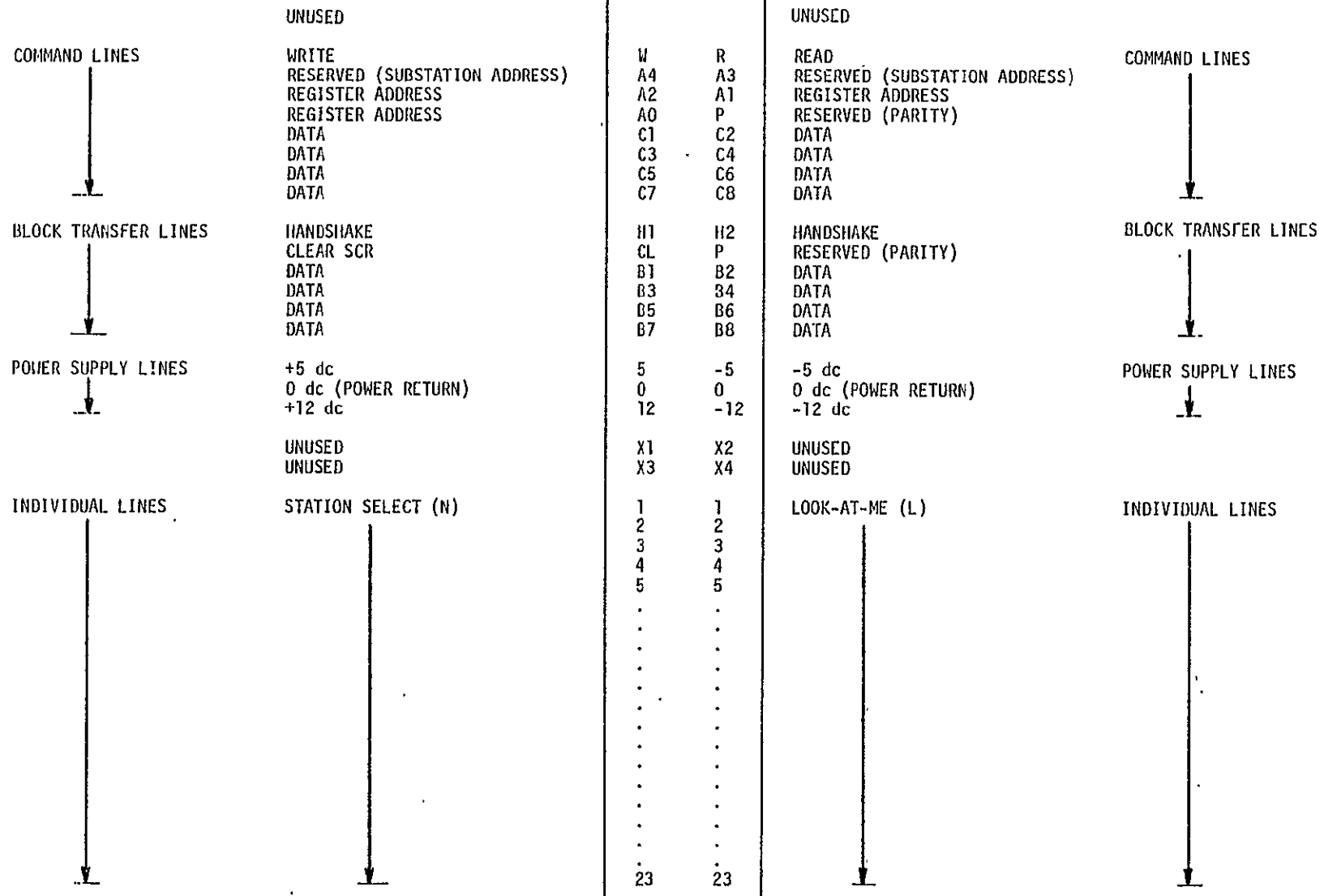


FIGURE 3-4. CONTACT ASSIGNMENTS FOR CU (STATION 24)

- A microcomputer that performs PDCC executive functions
- The logic for decoding the microcomputer's address lines and setting the state of the select lines to each station
- A register to buffer the LAM lines connecting the CU to each CP station
- Output and input buffers for controlling devices in the CU and for sensing their status
- An interface for regulating the movement of data between the PDCC and the host computer.

Figure 3-5 is a schematic block diagram of the control unit, showing the items in the above list and the interconnecting buses. Subsequent paragraphs present the design requirements for the devices in the CU.

### 3.3.1 CU Microcomputer

The CU microcomputer (Figure 3-6) consists of an Intel 8080A microprocessor, or equivalent, and the associated integrated circuits needed to construct a computer (e.g., clock, system controller, address decoders, bus drivers, and memory). It includes a Read-Only Memory (ROM) for program storage and system constants and a Random Access Memory (RAM) for dynamic storage. The capacity of these memory units is not specified because the program size for the CU microcomputer depends on the characteristics of the host computer's I/O channel, among other things. If the PDCC initiates and supervises data transfer, the executive program is larger than would be the case if the host computer's I/O channel had this role.

### 3.3.2 Station Select Unit

Address lines A<sub>5</sub> through A<sub>15</sub> of the CU microcomputer are decoded by the Station Select Unit. A memory reference to addresses above 65511 (decimal) sets the select line of a specific station according to the assignments shown in Figure 3-7, which includes a typical decoding circuit using the Intel 8205 as shown in Figure 3-4. If the address lines A<sub>10</sub> through A<sub>15</sub> are 1's, the address lines A<sub>5</sub> through A<sub>9</sub> are decoded to select

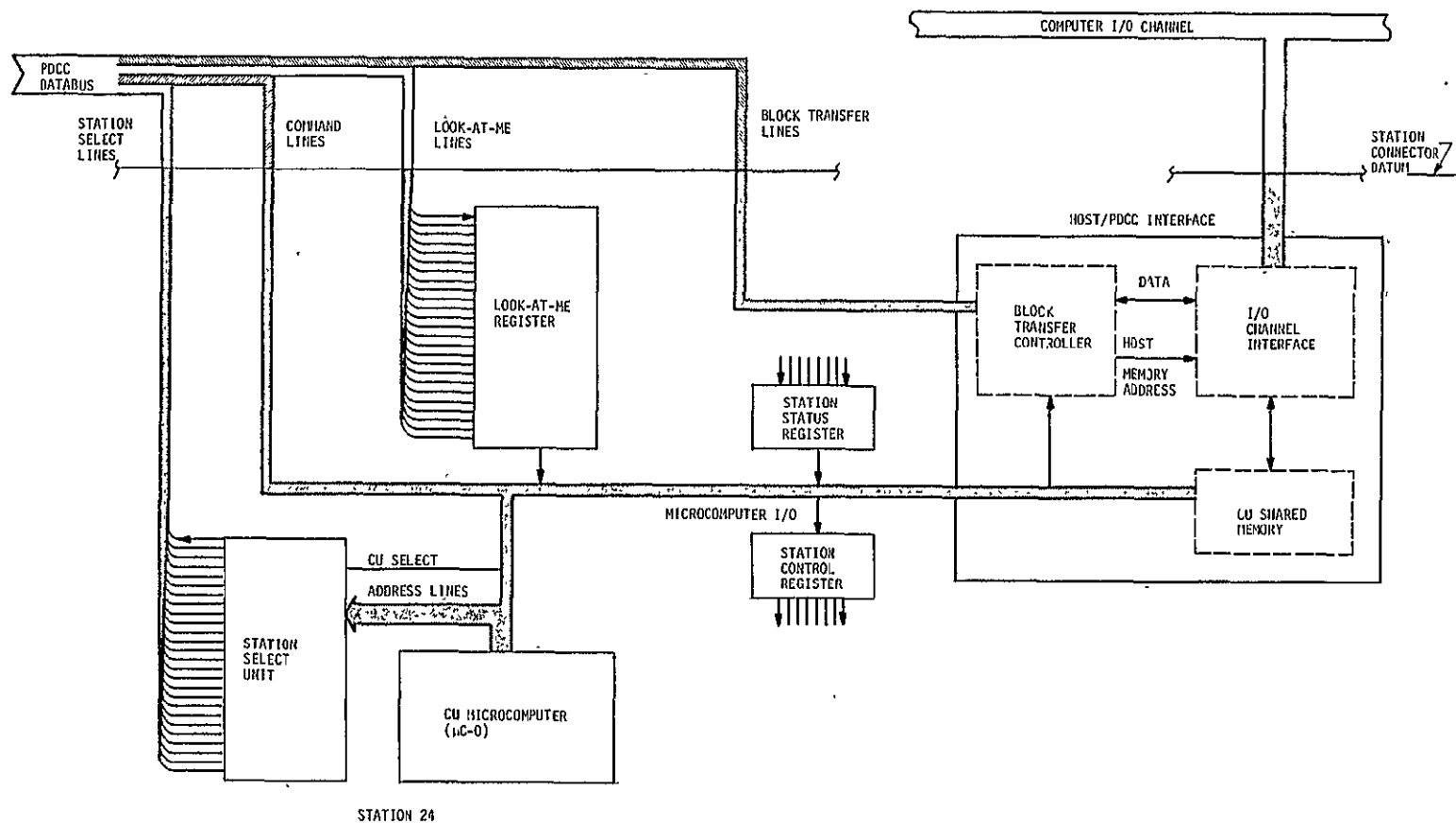


FIGURE 3-5. CONTROL UNIT (CU) SCHEMATIC BLOCK DIAGRAM

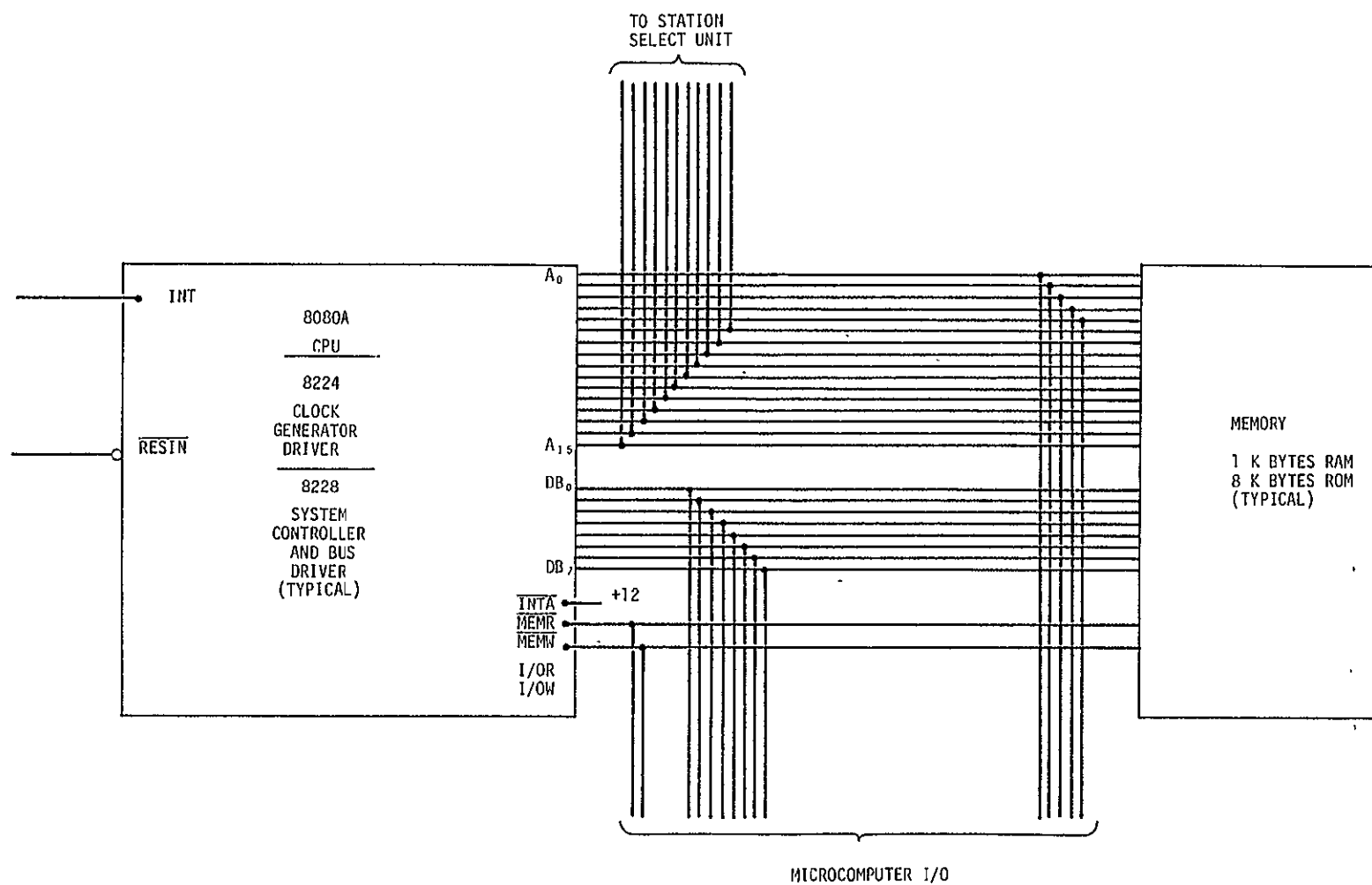


FIGURE 3-6. CU MICROCOMPUTER



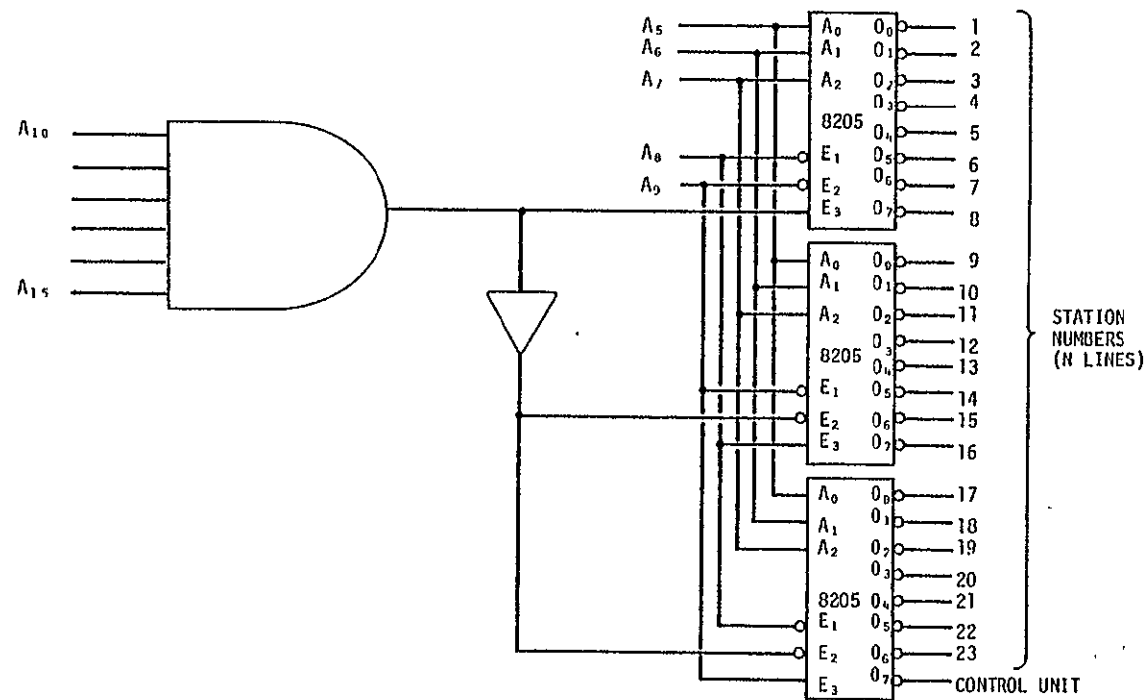


FIGURE 3-7. STATION SELECT UNIT

one of the 23 CP stations or to select the CU. The CU microcomputer performs I/O with devices in the selected unit.

### 3.3.3 Microcomputer I/O

The CU microcomputer has a memory-mapped I/O. Figure 3-8 shows the logic for selecting or controlling peripheral devices on the microcomputer and for generating the I/O read and write strobes. The assignment of some I/O bus lines for the host/PDCC interface is unspecified because register organization in the block transfer controller will depend on the word size of the host and on the features of a particular DMA channel. The CU microcomputer I/O has two components: one that connects to the devices in the CU (i.e., at Stations 24 and 25) and another that terminates on the plug of the PDCC Databus connector. The connections to the Databus are as follows:

<u>CU MICROCOMPUTER</u>	<u>DATABUS COMMAND LINES</u>
DB <sub>0</sub> - DB <sub>7</sub>	C <sub>1</sub> - C <sub>8</sub>
A <sub>0</sub> - A <sub>4</sub>	A <sub>0</sub> - A <sub>4</sub>
I/O W, I/O R	W, R

As a consequence of this arrangement, the command lines of the Databus are an extension of the CU microcomputer I/O. Devices in the CP stations (Stations 1 through 23) that connect to the Databus command lines are also peripherals to the microcomputer in the control station.

### 3.3.4 Look-at-Me (LAM) Register

The LAM register is a buffer for the 23 LAM lines that link the CU to each CP station. Logically, it is equivalent to three Intel 8212 chips operating as 8-bit input ports, as shown in Figure 3-9. The CU microcomputer reads the register to identify ports that have the LAM line set.

### 3.3.5 Control and Status Registers

The CU design makes provision for a control register that is equivalent to a single Intel 8212 operating as a gated output buffer

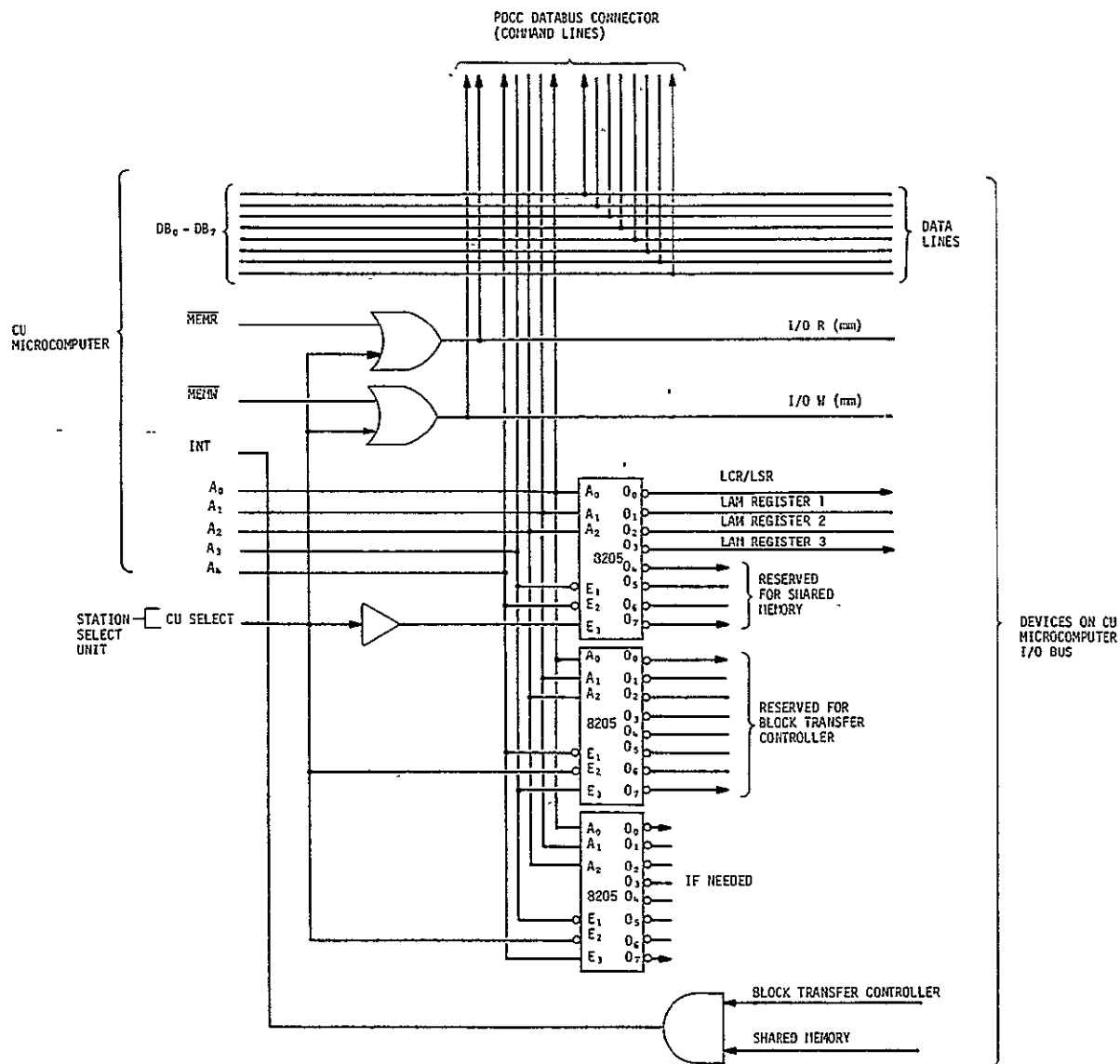


FIGURE 3-8. CU MICROCOMPUTER I/O

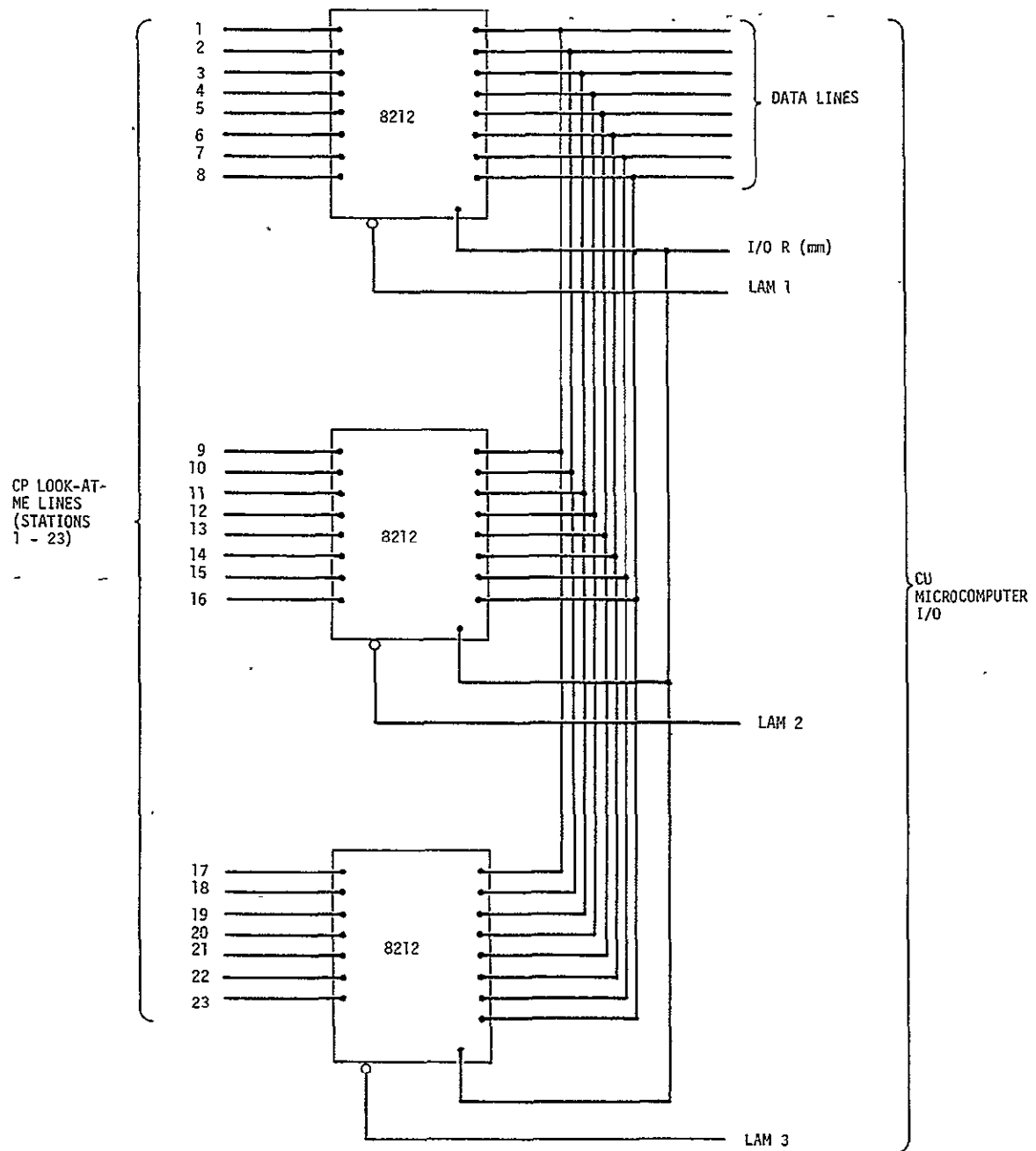


FIGURE 3-9. LOOK-AT-ME REGISTER

and a status register equivalent to an 8212 operating as an input port. Figure 3-10 is a representative configuration for these registers. Their use is dependent on the specific implementation of the host/PDCC interface. The assignment of the control and status lines in Figure 3-10 is for the interface arrangement shown in Figure 3-5.

### 3.3.6 Host/PDCC Interface

The host/PDCC interface is the part of a PDCC that must be specified separately for each distinct computer. It must provide the facilities for:

- Transferring word-oriented data between the CU microcomputer's I/O bus and the host computer's I/O channel
- Transferring blocks of data between the host computer's I/O channel and the block transfer lines in the PDCC Databus.

Figure 3-5 shows the preferred configuration, for use with direct memory access channels. It includes:

- A block transfer controller for
  - ▲ Generating memory reference addresses to store data in or fetch data from the host computer's memory
  - ▲ Synchronizing data movement on the block transfer lines of the PDCC Databus
- A shared memory for temporarily storing
  - ▲ Instructions that the host computer sends to the CU microcomputer
  - ▲ Status information that the microcomputer sends to the host
- An I/O channel interface that performs assembly/disassembly of word-oriented data and compensates for electrical and timing differences between the PDCC and the host computer's I/O channel.

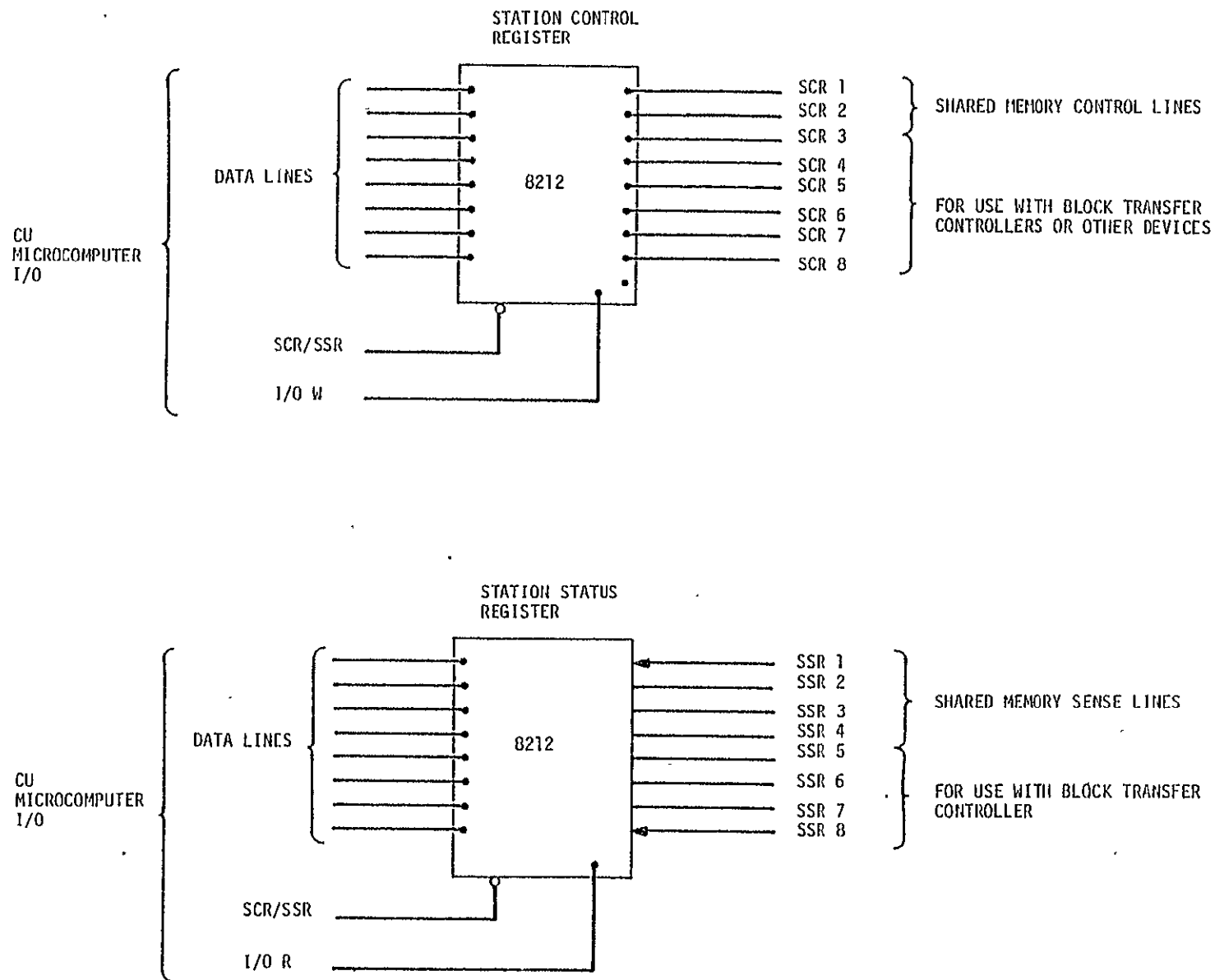


FIGURE 3-10. STATION CONTROL REGISTER (SCR) AND STATION STATUS REGISTER (SSR) FOR CU

This configuration of the host/PDCC interface would be effective for the PDP-11 UNIBUS, the Interdata 8/32's EDMA Bus, and the Varian I/O bus (self-contained BTC). -- among many others.

The block transfer controller in Figure 3-5 includes the registers for storing the starting and ending addresses of the memory area in the host computer from which data are to be read or to which data are to be written. Beginning at the starting address, it proceeds by cycle stealing to transfer data from/to successive memory locations until the ending address is reached. During the transfer of data, the block transfer controller must manipulate the handshake lines (H1, H2) in the PDCC Databus that cause a port to receive or send successive bytes of data.

Typically, the shared memory shown in Figure 3-5 is a pair of first-in, first-out (FIFO) storage devices. The CU microcomputer sends status information to the host computer by writing into the input of one and receives control instructions from the other by reading its output. The CU shared memory is similar to the CP shared memory (Figure 3-11) but has significantly less storage capacity.

#### 3.4 COMMUNICATION PORT (CP)

Each communication port is a circuit on a plug-in board that occupies one of the CP stations (Stations 1 through 23). It consists of the following units:

- A microcomputer that performs line control and terminal control functions for a communication line
- Command registers for exchanging control and status information with the control unit
- Status and control registers that are accessible only to the control unit
- Status and control registers that are accessible only to the CP's microcomputer

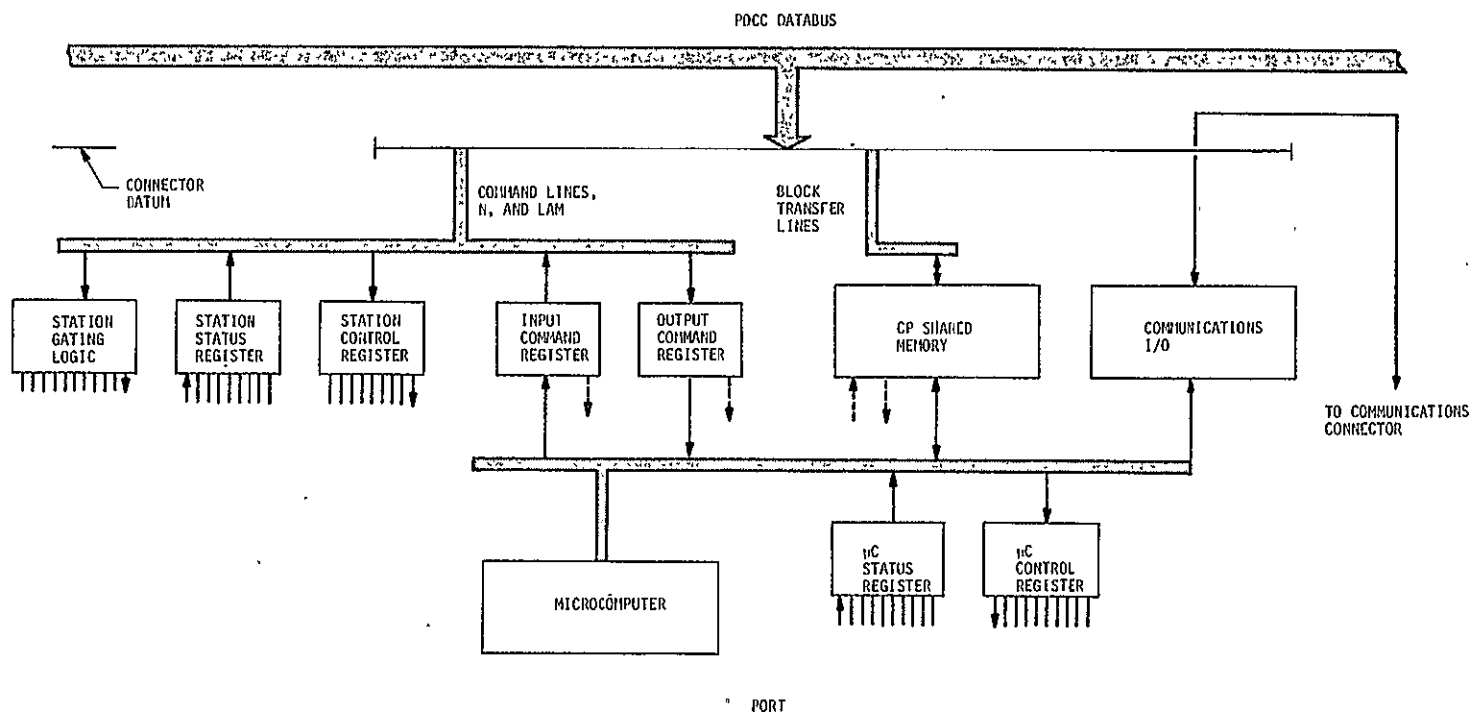


FIGURE 3-11. PLUG-IN BOARD AT STATION N



- A shared memory unit for buffer storage of data blocks that are segments of messages being sent to or received by the host computer
- A communications interface that provides a means of transferring data and control signals between the data communications equipment (e.g., modem) and the microcomputer's I/O bus
- The microcomputer's I/O
- Gating logic for register selection.

Figure 3-11 is a schematic block diagram of a communication port showing these devices. Subsequent paragraphs of this section describe the requirements of each. To make allowance for recent and future increases in circuit density, the PDCC design requirements provide for up to four communication ports at each station. The implications are shown in the diagrams of the units affected.

#### 3.4.1 Gating Logic

The function of the gating logic is to recognize the state of the station select line and to decode the register address lines (A<sub>0</sub> through A<sub>4</sub>) of the PDCC Databus, if L is set. The outputs of the gating logic connect to the select and/or strobe inputs of the ICR, OCR, SCR, and SSR. Figure 3-12 specifies the decoding requirement in terms of the Intel 8205 Binary Decoder.

To benefit from present and future increases in the level of integration in electronic circuits, the design requirements make provision for more than one port at a station (up to four). The gating logic shows the address assignments for the additional CPs.

#### 3.4.2 Station Control Register (SCR)

The SCR is a device on the PDCC Databus. As shown in Figure 3-13 (upper half), it is an 8-bit latched output port with its input data lines connected to the station's Databus connector. The purpose of this register is to set the mode of the shared memory (receive, send, or off). The possibilities are:

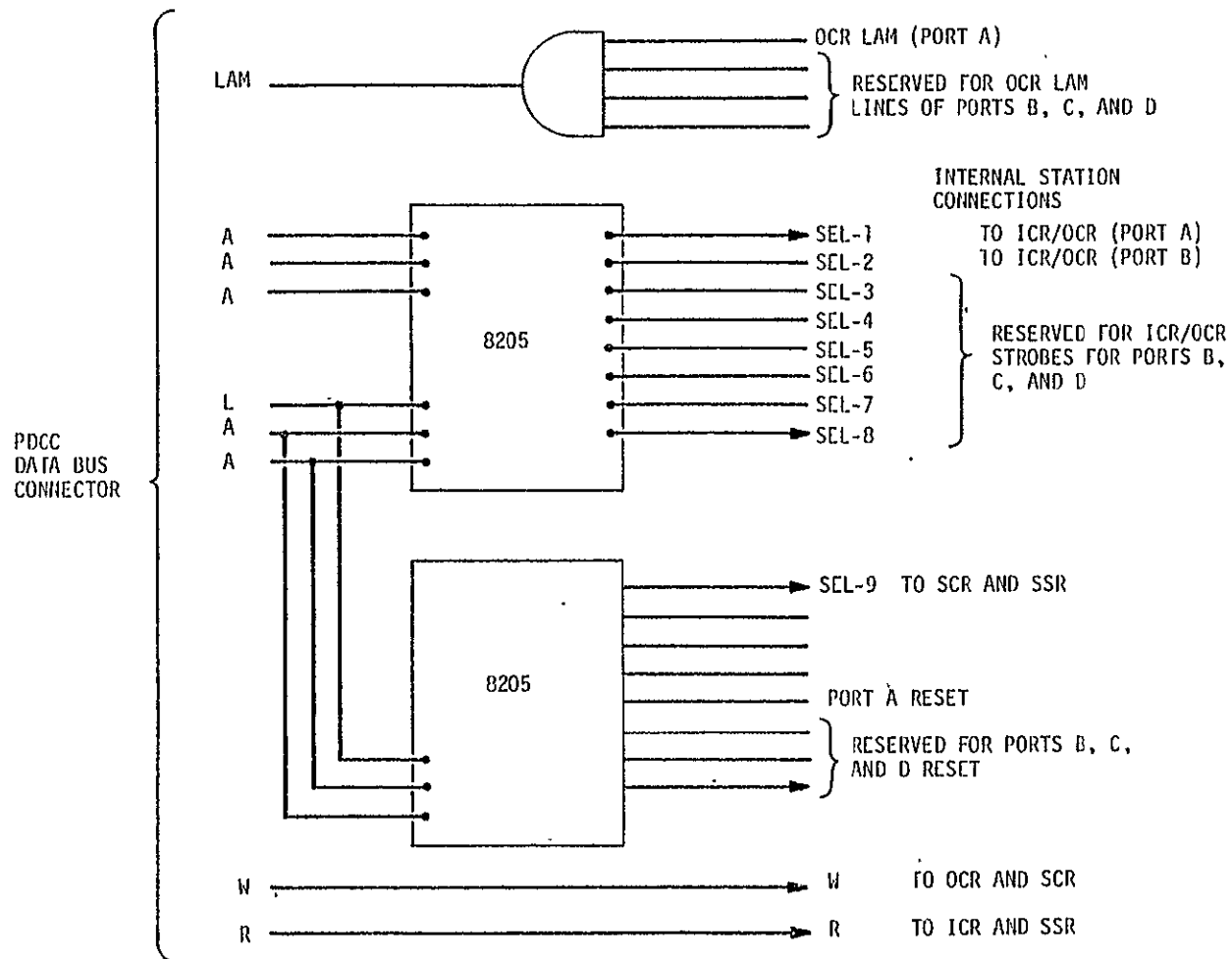


FIGURE 3-12. GATING LOGIC AT EACH STATION

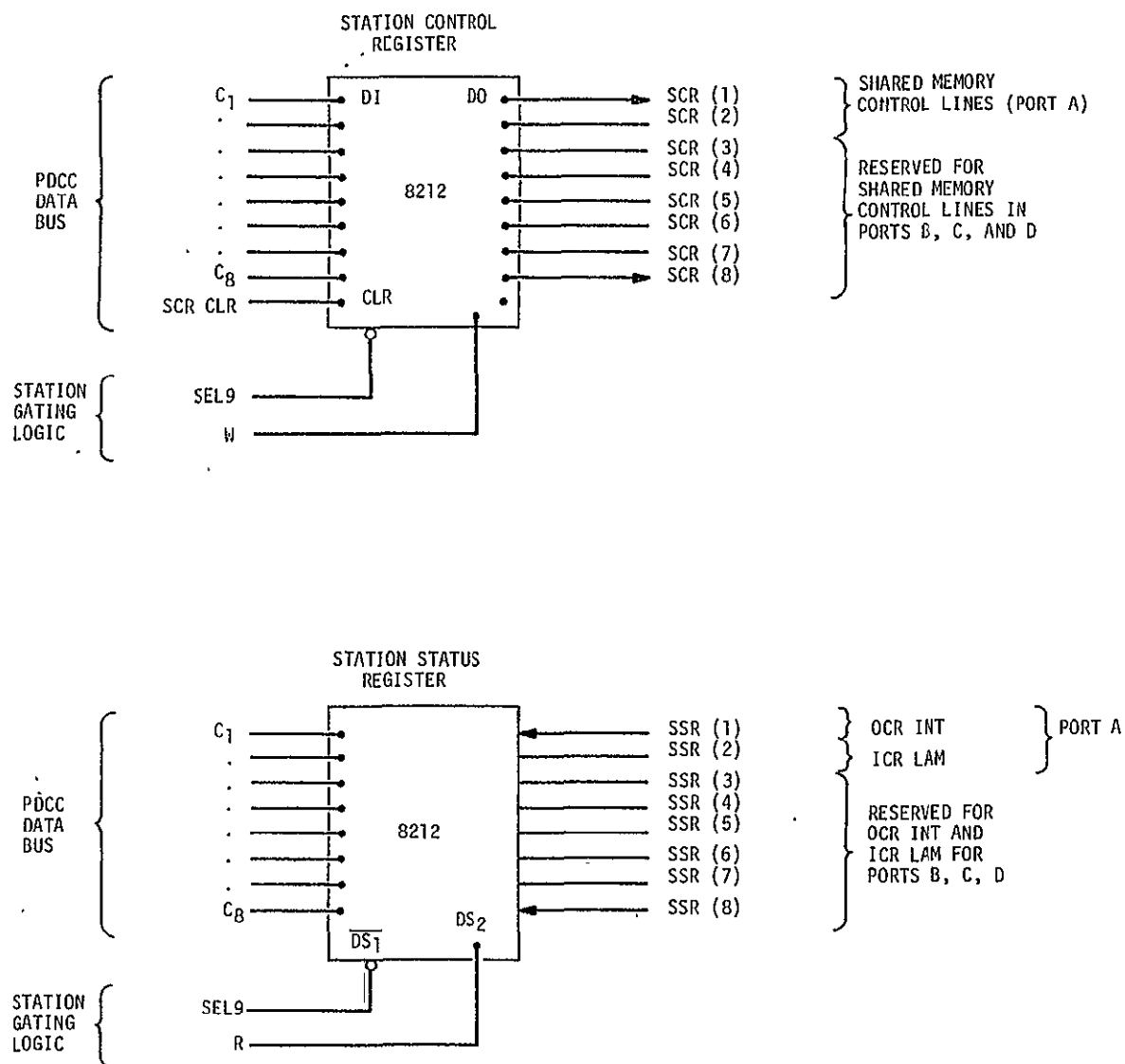


FIGURE 3-13. STATION CONTROL REGISTER (SCR) AND STATION STATUS REGISTER (SSR)

<u>BIT 1</u>	<u>BIT 2</u>	<u>SHARED MEMORY MODE</u>
0	0	Off
1	0	Receive from Databus
0	1	Send to Databus
1	1	Illegal

As with the gating logic, the SCR is specified with the provision for having up to four CPs at a station. The Databus control line SCR CLR connects to the clear contact of the output port.

#### 3.4.3 Station Status Register (SSR)

The lower half of Figure 3-13 exhibits the requirements for the SSR, using the Intel 8212 I/O port as a model. The latches within the 8212 continually track the input data lines; consequently, the status of these lines can be read directly from the PDCC Databus at any time by selecting the SSR as the I/O device.

By reading the SSR via the Databus, the control unit can determine whether the output command register is busy. When the CU writes a word into the OCR, the OCR INT line is set and remains set until the CP microcomputer reads this word. The SSR also shows the state of the LAM line, which is set when the CP microcomputer writes a word into the input command register.

The requirement for the SSR makes provision for up to four ports at a station.

#### 3.4.4 CP Microcomputer

The port's microcomputer consists of:

- An Intel 8080 microprocessor or equivalent
- The integrated circuits for system control
- A PROM or a RAM for program storage
- A RAM for data storage
- The integrated circuits for driving and controlling its I/O bus.

Figure 3-14 shows a representative configuration for the microcomputer. New devices that are now entering the marketplace provide a higher level of integration than those shown in this figure. Shortly, the chip count will be three or less for implementing the microcomputer with 1K bytes of PROM and 64 bytes of RAM (typically the memory needed for a TTY equivalent asynchronous terminal).

#### 3.4.5. Microcomputer I/O

The CP microcomputer has a memory-mapped I/O as shown in Figure 3-15. If bit 15 of the microcomputer's address bus is set, the low-order address lines ( $A_0$  through  $A_3$ ) are decoded to generate control pulses for selecting/clearing I/O devices. Of course, a specific implementation of the I/O can assign more memory address space to program and data by substituting a combination of high-order address lines for the single  $A_{15}$  line.

The I/O read and write strobes are generated in the usual way from  $\overline{\text{MEMR}}$  and  $\overline{\text{MEMW}}$  signals from the microcomputer. A common interrupt is obtained by logically (or) combining the INT line of the OCR and a line that indicates a service need for the communications interface.

#### 3.4.6 Input Command Register (ICR)

The ICR is a write-only device on the microcomputer's bus and a read-only device on the PDCC Databus. It is a 16-bit register with each of the two (8-bit) bytes being separately gated into or strobed out of the unit. Figure 3-16 is a schematic block diagram of a typical ICR, using the Intel 8212. The register must set an interrupt line when a byte is written into the most significant half of the register (ICR-2). When ICR-2 is read, the register must reset the interrupt line. The interrupt line is used to set the CP's LAM line.

#### 3.4.7 Output Command Register (OCR)

Figure 3-17 exhibits the requirements for the OCR using the Intel 8212 I/O port as a model. This is a 16-bit register that is a write-only device on the PDCC Databus and a read-only device on the CP microcomputer.

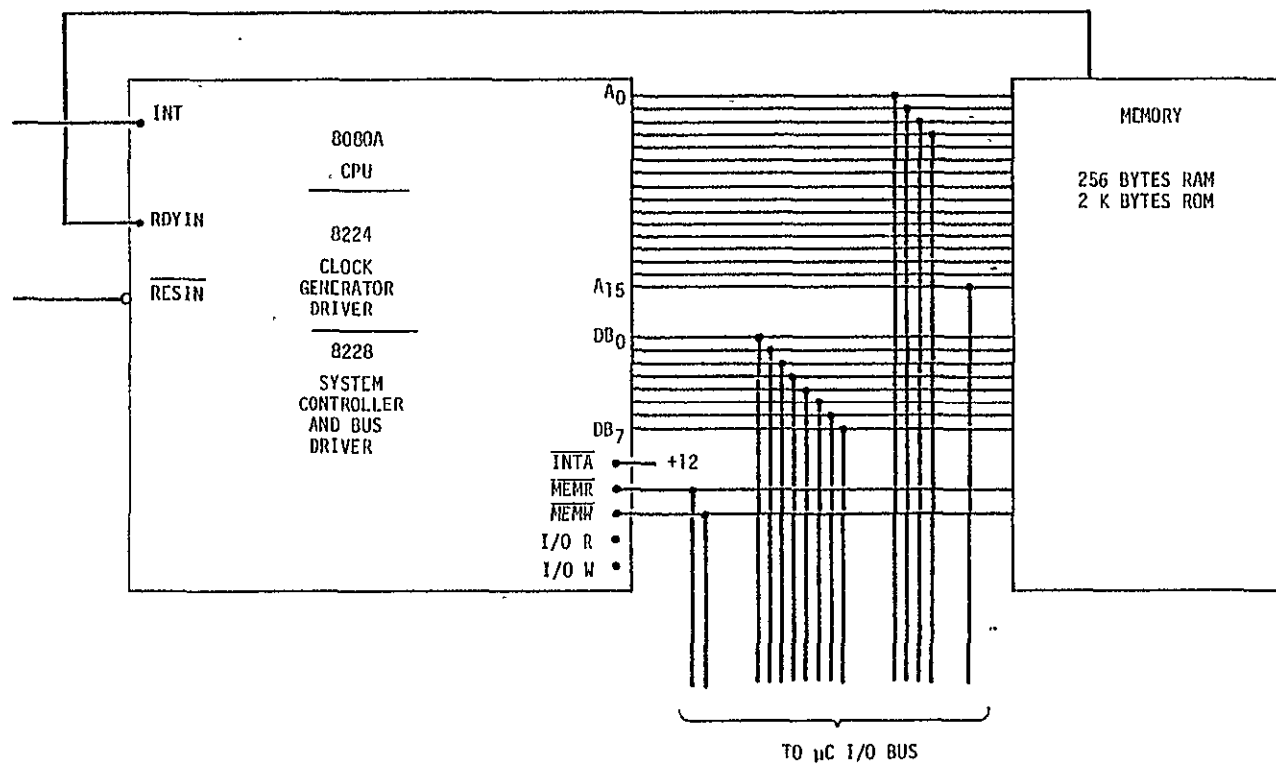


FIGURE 3-14. CP MICROCOMPUTER

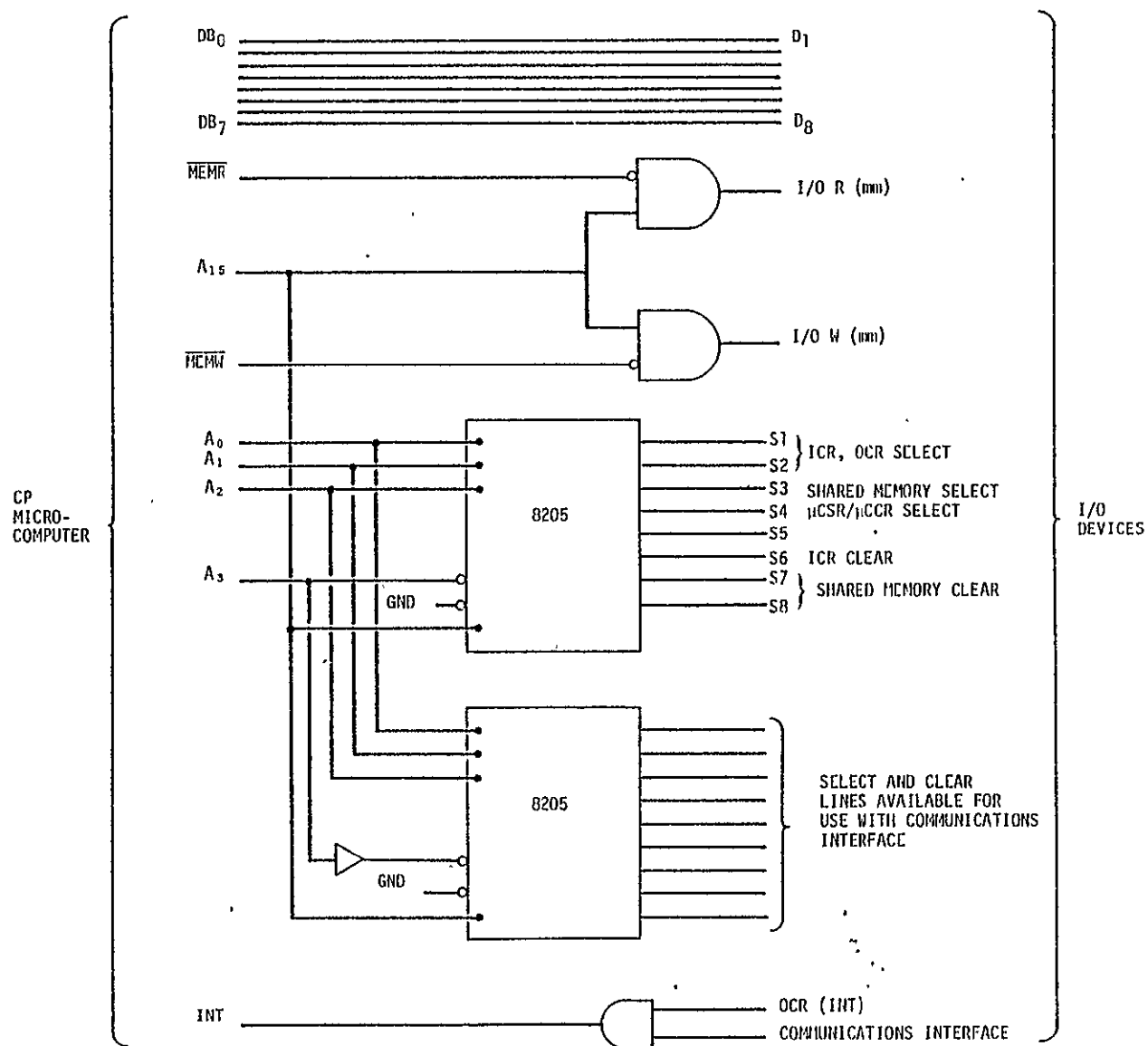


FIGURE 3-15. CP MICROCOMPUTER I/O

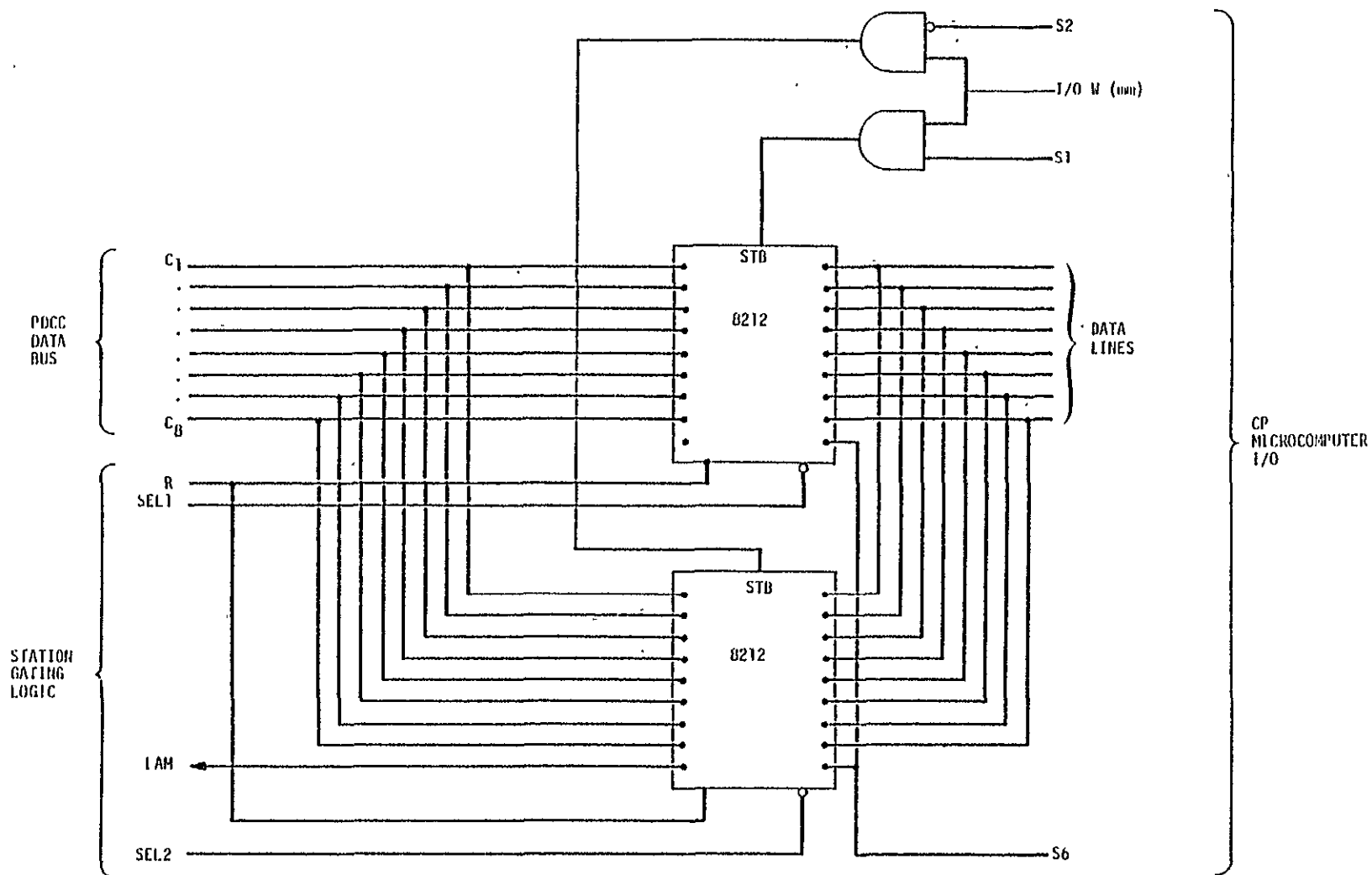


FIGURE 3-16. INPUT COMMAND REGISTER (ICR)



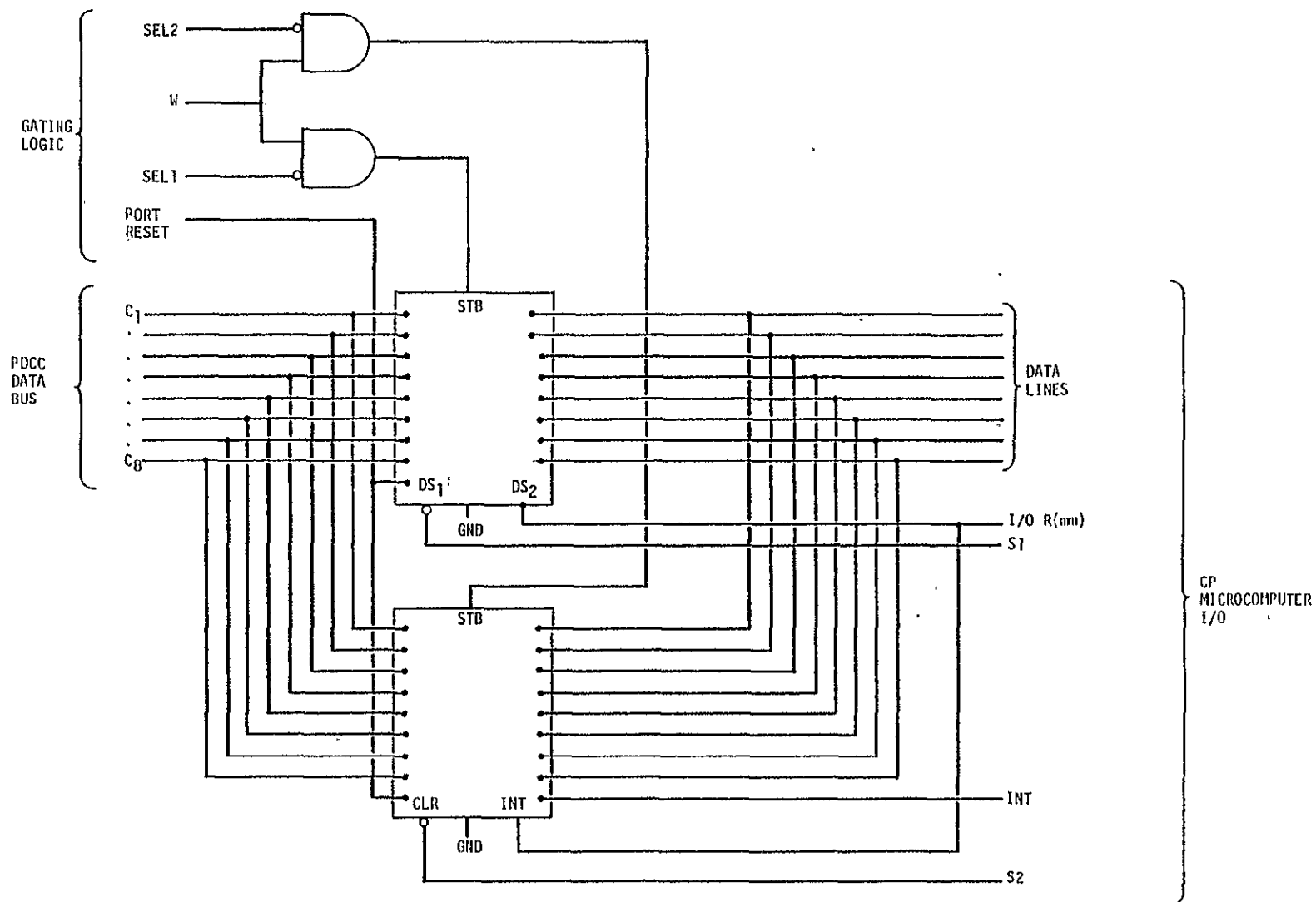


FIGURE 3-17. OUTPUT COMMAND REGISTER (OCR)

The most significant and the least significant 8-bit bytes of the OCR are written and read independently. Strobing data into OCR-2, the most significant half of the register, sets the INT line that is used to interrupt the microcomputer. Reading OCR-2 resets the INT line.

#### 3.4.8 CP Microcomputer Control and Status Registers

The control and status registers are conventional input and latched output ports on the microcomputer's I/O. Figure 3-18 shows the assignment of status lines into the status register. The control register is available for setting mode and control lines (i.e., baud rate) associated with the communications interface.

#### 3.4.9 Shared Memory

The CP shared memory consists of two first-in, first-out (FIFO) storage units. Each unit has separate input and output ports. The input port of one unit connects to the PDCC Databus connector. Its output port is an I/O device on the CP microcomputer. The input port of the other unit is an I/O device on the CP microcomputer; its output port connects to the PDCC Databus connector. Consequently, the shared memory is two FIFO channels. One is a buffer for communication data arriving at the CP from the host computer via the PDCC Databus. The other is a buffer for data received by the port on the communication line and destined for the host computer via the PDCC Databus.

Figure 3-19 is the schematic block diagram of a shared memory, based on the Fairchild 3341A FIFO memory chip and the Intel 8212. It has a capacity of 128 bytes in each channel. The 3341A chips are interfaced to the CP microcomputer I/O and to the PDCC Databus with the 8212 I/O ports. The device configuration in Figure 3-19 was chosen for its illustrative features. It obviously does not represent a design solution that minimizes chip count.

The mode of the shared memory is determined by the state of the two lines labeled SCR(1) and SCR(2). If both are off, the shared memory is disabled from the interchanging data with the PDCC Databus. An "on"

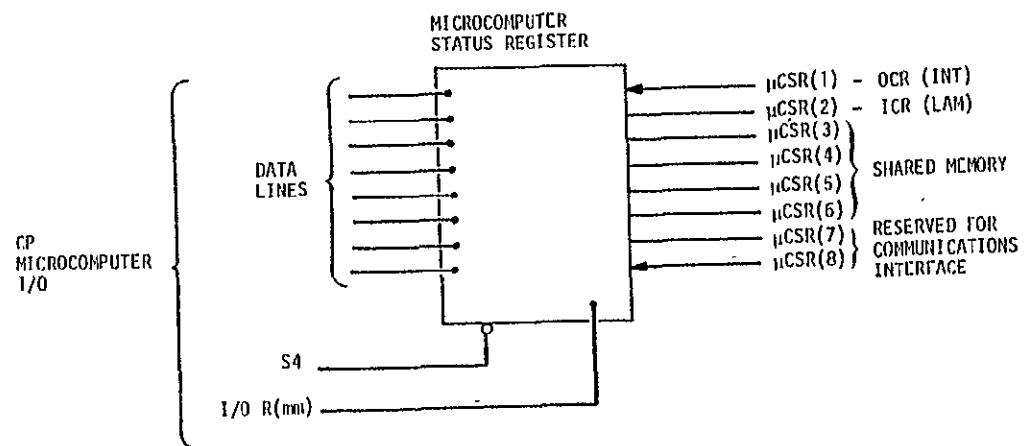
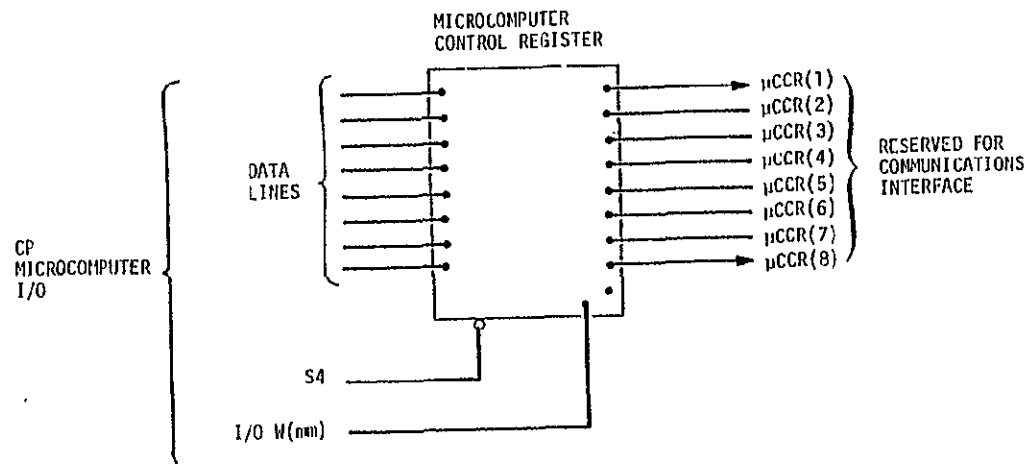


FIGURE 3-18.  $\mu\text{C}$  CONTROL REGISTER AND  $\mu\text{C}$  STATUS REGISTER IN CP

condition for SCR(1) enables the shared memory to receive data from the Databus. SCR(2) in the "on" condition enables the sending of data to the Databus.

The shared memory is an I/O device on the CP microcomputer. Reading this device retrieves the front byte from the upper channel in Figure 3-19. Writing to the shared memory enters a byte into the lower channel.

#### 3.4.10 Communications Interface

Design requirements for the communications interface are not specified here. They depend on the line protocol of the data communications equipment and on the standards met by the communications connector.

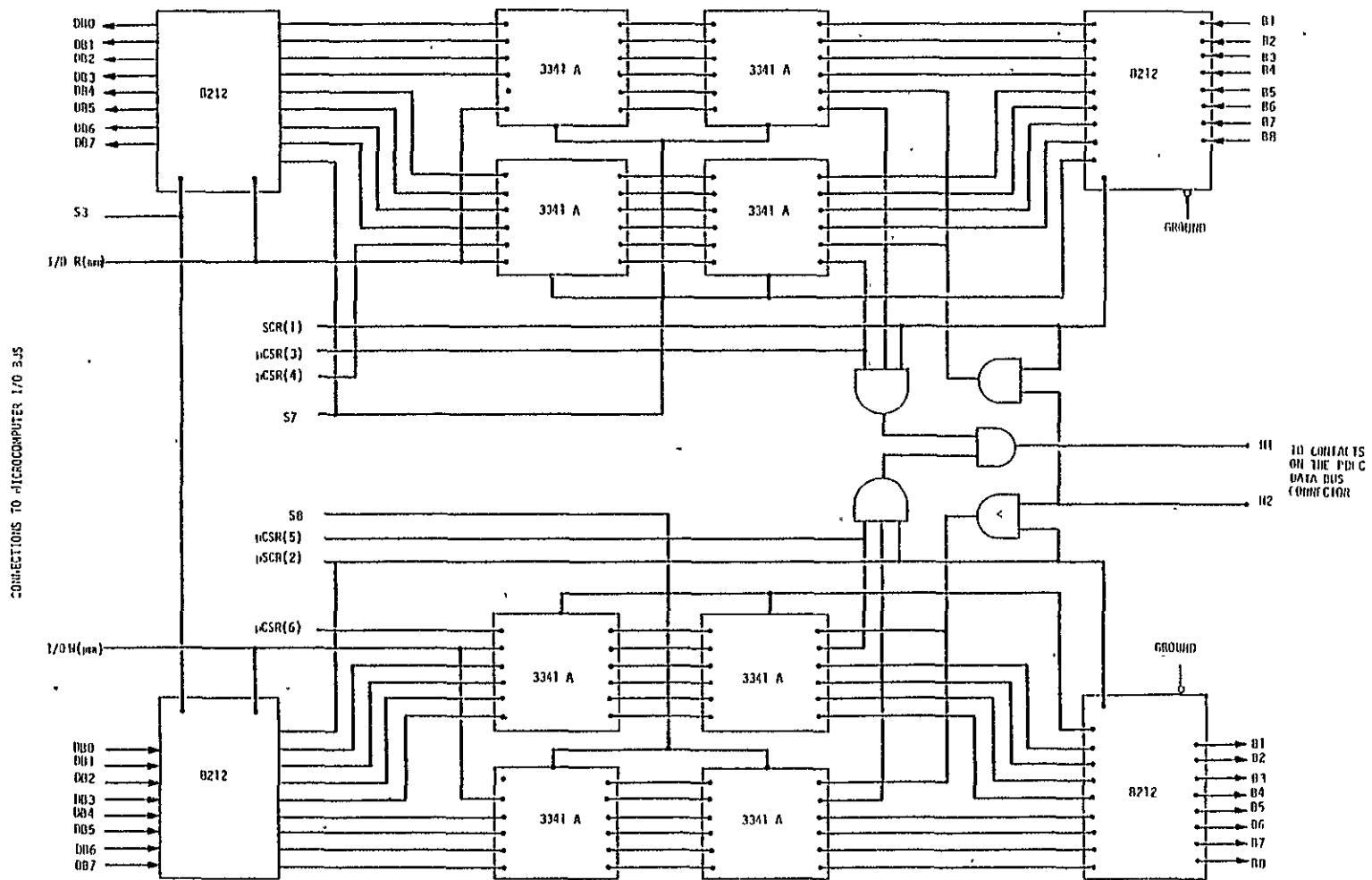


FIGURE 3-19. CP SHARED MEMORY

## 4. OPERATION OF THE PDCC

Functionally, the PDCC and the host computer interchange data over two paths. One path is for the transfer of control and status information between the PDCC control unit and the host computer and between the control unit and the control ports. The other path is the route for moving communications data (i.e., the contents of communication messages) between the host computer and the communication ports. The first is oriented toward the interchange of information in units of one command word. The second is a facility for transferring data blocks of 60 bytes each. The two paths appear physically in the PDCC Databus as: 1) the command lines and 2) the block transfer lines.

Many operations performed by an individual communication port are the consequence of the line protocol it supports and the terminal control it provides. Further, the performance of the control unit depends in part on the characteristics of the host computer, its I/O channel, and the implementation choices made by the designer of a specific PDCC. The subject of this section, however, is the operating principles that are common to all PDCCs. Subsequent sections discuss the performance in terms of the two data paths and the protocol for transferring command information and communication messages between the computers in the distributed network formed by the host computer, the control unit, and the CPs. To present a complete discussion, assumptions occasionally need to be made about the nature of the data transfer between the PDCC and the host computer. As in Section 3.3, the model used here is: The PDCC connects to a programmed I/O bus for interchange of command words and to the direct memory access facility for interchange of data blocks.

### 4.1 INTERCHANGE OF CONTROL AND STATUS INFORMATION

The host computer and the PDCC interchange control and status information through the CU shared memory. The control unit and the communication ports interchange control and status information through the command registers in each CP. Figure 4-1 shows this network for managing operation of the PDCC.

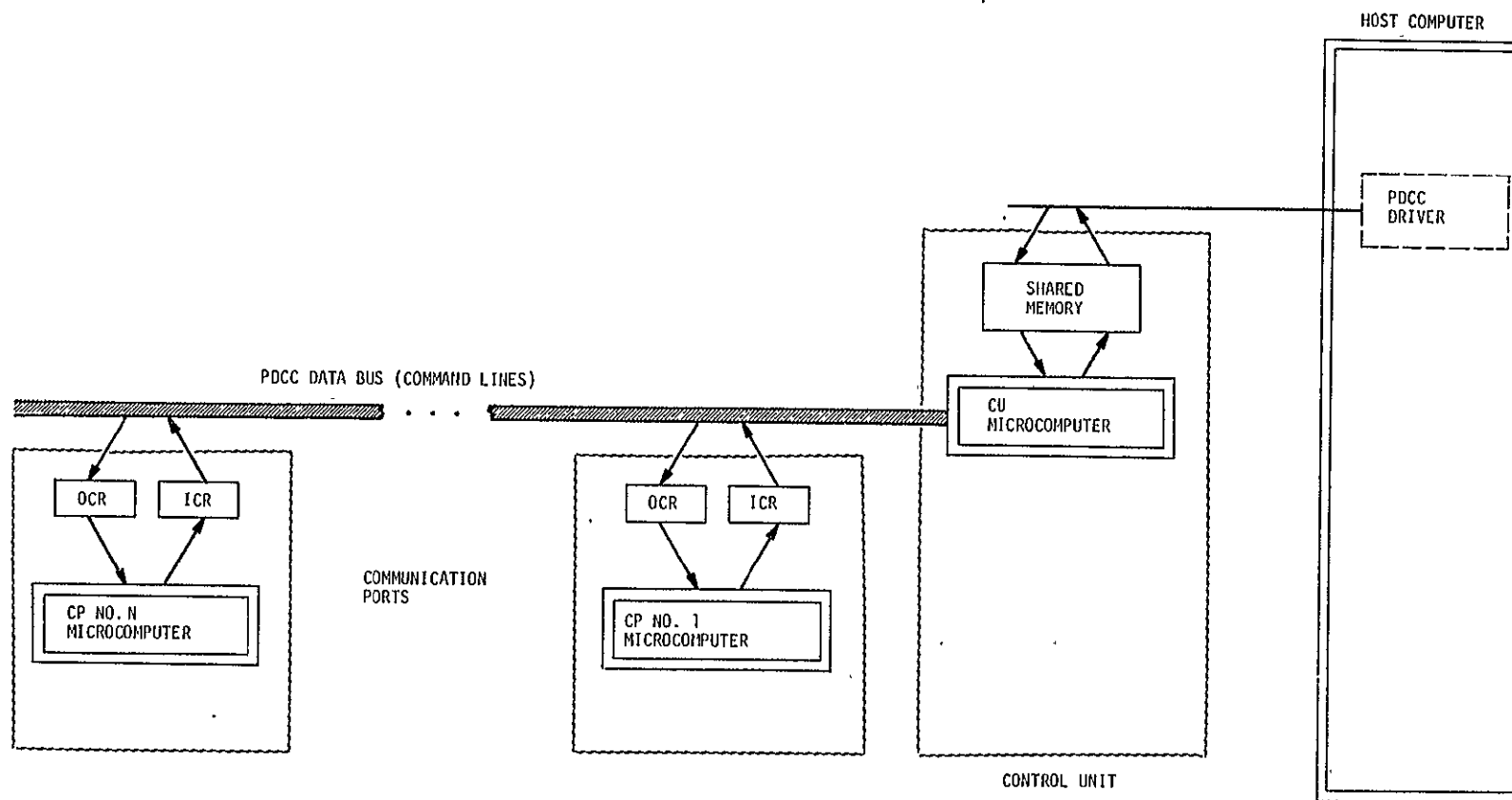
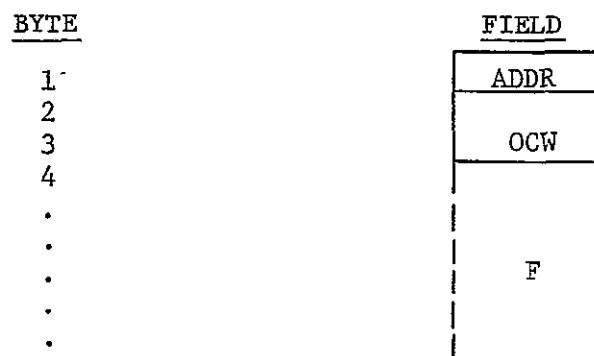


FIGURE 4-1. FUNCTIONAL DIAGRAM OF COMMAND INTERCHANGE

The PDCC driver sends a directive to the PDCC by executing write instructions that transfer an output command into the output channel of the CU's shared memory. Conversely, the driver receives a status report from the PDCC by executing read instructions that retrieve an input command from the input channel of the CU's FIFO memory facility. In response to the output commands it receives from the host computer, the control unit sends control information to a CP by writing an output command word (OCW) into the port's output command register (OCR). In the reverse direction, the CU receives status from a communication port by reading an input command word (ICW) from the port's input command register (ICR); this information is placed in an input command to the host computer.

The PDCC control unit will have some characteristics that depend on properties of the host computer. Consequently, the design requirements presented here (which are computer-independent) cannot completely specify the attributes necessary for the interchange of commands between the host computer and the PDCC. For instance, the control unit may perform operations that are required of all peripherals on the host computer's I/O channel but are unrelated to performance. Nevertheless, subsequent paragraphs describe a representative set of output commands and input commands for the host computer to use in transmitting directives to the PDCC and receiving status reports from it. They constitute design requirements to the extent that a PDCC implementation must include commands equivalent to these.

The directives sent by the PDCC driver to the CU (Figure 4-1) include the following fields:

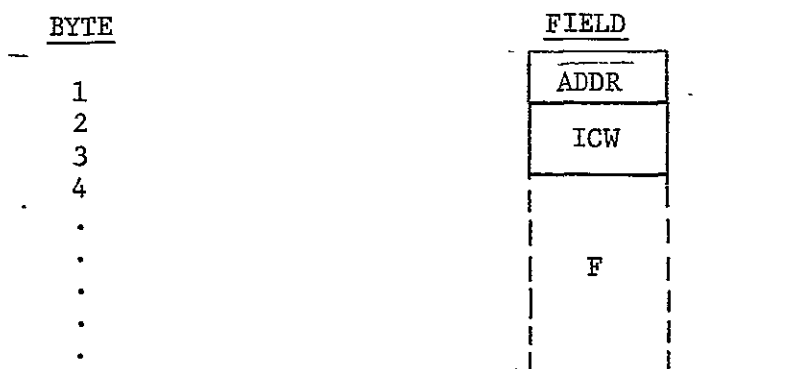




where ADDR specifies the CP address (1-92) and OCW is the output command word that the control unit sends to the selected port. Most output commands include only these two fields; for them, the CU performs only a multiplexer function. They are a vehicle for application programs to send control variables to a communication port (say, through a function call to the PDCC driver).

The CU does retain information from one group of commands: those that relate to executive functions of the PDCC. The executive group includes the directives to open and close a specified port and to send or receive a communication message. Some of these commands include additional fields to provide the CU with information about the location and size of data buffers in the host computer. Whether these fields are associated with the open command or with send/receive commands depends on the way I/O buffers are managed in the host computer. The information in executive commands is retained by the CU; however, the OCW is also forwarded to the specified port.

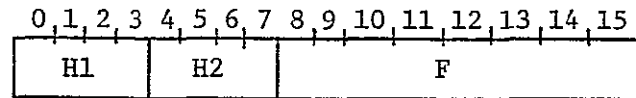
In a similar fashion, the status reports from the CU to the PDCC driver have the following form:



where ADDR is the address of the CP originating the report and ICW is the input command word read from the port. Most input commands include only the ADDR and ICW fields. The exceptions are the responses to the executive group of output commands and the executive status reports. Some of these may require additional fields (F) that are peculiar to each PDCC implementation.

#### 4.1.1.1 CU/CP Command Words

The command word (OCW/ICW) is the fundamental unit for interchange of control and status information in the PDCC. It consists of 16 bits in the following format:



where H1 and H2 represent hexadecimal characters and F can represent a binary number (0 to 255), an 8-bit coded character, or a set of eight control or condition flags, depending on the type of command.

The commands that the CU and the CPs use to interchange control and status information are arranged into 16 groups. These groups are indexed by the character H1. The other characters, H2, specifies a specific action or identifies the source/destination within the CP of the data in field F.

Two of the 16 command groups are standard for all PDCC implementations. The executive group (H1 = 0) includes the commands to open and close a communications port and to send/receive communication messages. CPs respond to these output commands with specific input commands that indicate completion of operations. The block transfer group (H1 = F) is a set of reserved commands that the CU and CP use to coordinate the transfer of communications data between the CP shared memory and the host computer. (It is illegal to have H1 = F in CU-to-host-computer command interchange.) The other command groups provide the means for an application program to manage CP operations that are related to specific line protocols (e.g., baud rate selection) and particular terminal control functions (e.g., triggering a terminal startup sequence automatically generated by the CP microcomputer).

The specification of the executive command group is crucially important to the PDCC concept. They must have a universal interpretation in all implementations because the interchangeability of CPs among

different PDCCs depends on this property. Subsequent paragraphs of this section present a tentative initial version of the executive group, but the final selection is an appropriate action for a NASA working group (similar to an ANSI committee).

Table 4-1 lists nine OCWs that are candidates for the executive group and defines their functions. The open command directs a CP to exit the unique standby state and enter a normal mode of operation; the close command returns it to standby. There are four commands related to sending and receiving communication messages. The send command instructs a port to transmit one message, or a sequence of messages, using data from the shared memory. Conversely, the receive command enables the port's receiving function for reception of one command, or a sequence of commands. Because the send and receive commands can be used for initiating a sequence of messages, appropriate means must be available for notifying a CP that the sequence has ended. The two codes H2 = 5 and H2 = 6 are available for this purpose.

The control command directs a CP to take the action specified by the modifier field, this byte being supplied by the application program (via the PDCC driver). The sense command solicits a status report from the CP, identifying the report with the modifier field.

Table 4-2 lists the ICWs that are recommended for inclusion in the executive group. Most of them have a direct relationship with the corresponding entry in Table 4-1. One exception is the ready/acknowledge command that a CP uses to signify acceptance of an executive OCW. Another is the error command that reports system-level errors (e.g., shared memory overflow, illegal command for the CP).

Upon sending/receiving the last character of a message, the CP notifies the control unit with an EOM command. If the port supports multiple message sequences, it also uses an EOF command to notify the CP that the last message has been sent/received. The status ICW responds to the sense OCW. The control ICW can continue a command sequence begun by the corresponding OCW.

TABLE 4-1. EXECUTIVE GROUP OF THE OUTPUT COMMAND WORDS (H1 = 0)

H2	MNEMONIC	FUNCTION	USE OF MODIFIER FIELD (F)
0	NO OP	Used to clear OCR with no actions taken	F = 00
1	OPEN	Instructs CP to prepare for operation in the mode specified by F	Mode selection for multipurpose CPs; contents supplied to PDCC driver by application program
2	CLOSE	Instructs CP to a wait state preparatory to receiving an open command	Not used
3	SEND MSG	Notifies CP that PDCC has received a send message command from PDCC driver	Available for application program to provide message sequence number
4	REC MSG	Instructs CP to accept next incoming message from communication line	Available for application program to provide message sequence number
5	STOP SEND	Instructs CP to stop sending messages	Not used
6	STOP REC	Instructs CP to stop accepting incoming messages	Not used
7	SENSE	Instructs CP to return status	Available for application program to identify status source
8	CTRL	Instructs CP to accept the F field as a control code	Application program provides control code
9 ↓ 15	Unspecified		

TABLE 4-2. EXECUTIVE GROUP OF THE INPUT COMMAND WORDS

H2	MNEMONIC	FUNCTION	USE OF MODIFIER FIELD (F)
0	NO OP	Used to clear ICR; no actions required	F = 00
1	READY/ACK	Notifies CU that executive command has been executed/accepted	Contains the H2 code of the OCW that elicited this response
2	ERROR	Notifies CU of a system error	Error code
3	EOM/SEND	Notifies CU that last character of the message has been transmitted	Available for returning message sequence number or other data to application program
4	EOM/REC	Notifies CU that last character of the message has been received	Available for returning message sequence number or other data to application program
5	EOF/SEND	Notifies CU that last character of last message has been transmitted	Available for returning message sequence number or other data to application program
6	EOF/REC	Notifies CU that last character of last message has been received	Available for returning message sequence number or other data to application program
7	STATUS	Returns status in response to sense OCW (H2 = 5)	Status condition flags
8	CTRL	Facility for continuing control sequence begun by a control OCW (H2 = 6)	Available for returning control code to application program
9 ↓ 15	Unassigned		

#### 4.1.2 CU Command Processing

All of the computers shown in Figure 4-1 operate asynchronously. Output commands and input commands build up in the CU shared memory; these queues are serviced by the CU microcomputer and the host computer, respectively, in accord with task priorities and loadings of the two systems. The CU writes OCWs to a port, as needed, subject only to the condition that the corresponding OCR is not busy (i.e., that the CP's microcomputer has read the previous OCW). A CP sends status reports to the control unit by writing an ICW into the ICR at any time that the ICR is not busy (i.e., that the CU microcomputer has read the previous ICW). The CU periodically reads its LAM register to identify CPs that have input command registers containing unread status information.

The control unit of the PDCC performs the hardware functions of a multiplexer and, in part, the software functions of a communications executive.

#### 4.2 BLOCK TRANSFER OF COMMUNICATION DATA

Figure 4-2 shows the PDCC block transfer facility. It is a means for moving data between the communication ports and the host computer. These data are the text of the messages being sent to the terminals (via the communications interface) or received from them. The data transfers occur in units of 60 bytes, or a smaller number if the message text would otherwise end in the middle of a block.

The PDCC driver in the host computer initiates the transfer of a message to a remote terminal by writing a command into the CU shared memory of the PDCC. Based on information in the command, the CU microcomputer initializes the block transfer controller and connects the appropriate port to the block transfer lines in the Databus. The facility is set up for transferring the first 64 bytes of the message. The block transfer controller provides the timing and generates the handshake signals for reading the message segment into the CP's shared memory. Upon completion of the block transfer of data, the CU sends an output control

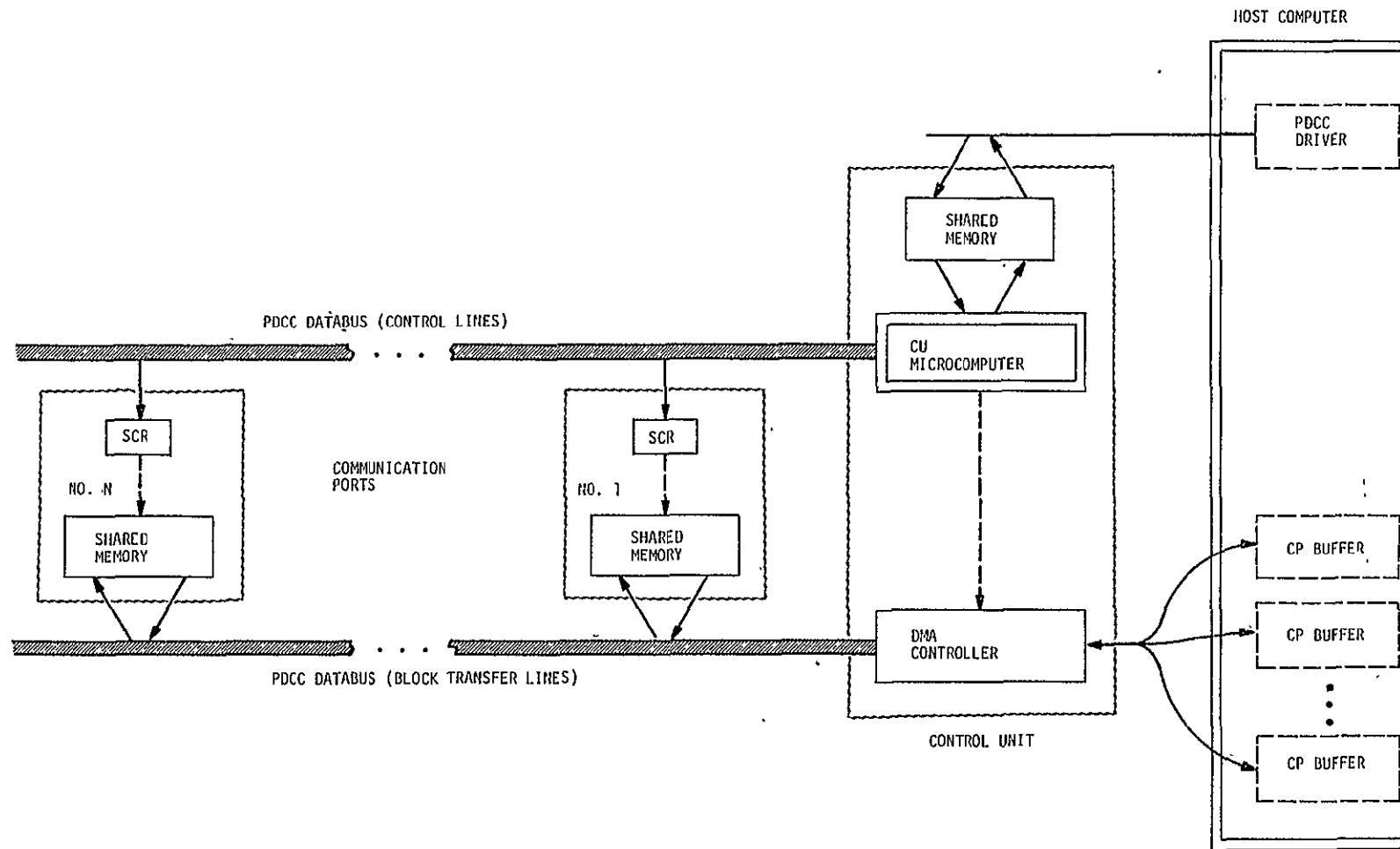


FIGURE 4-2. FUNCTIONAL DIAGRAM OF BLOCK TRANSFER OPERATION

word to the port, commanding the port to transmit the data block. A similar sequence is used for receiving messages.

The CU microcomputer manages the data interchange with the following operations:

- Preparation of a CP shared memory for loading/unloading data by writing the enabling code into the appropriate station control register
- Setup and initiation of the data transfer by writing control information into the block transfer controller
- Maintenance of control tables for multiplexing data blocks among the CPs.

In the configuration of Figure 4-2, the PDCC has a direct memory access to the host computer. Consequently, the CU manages the block transfer controller by supplying an address and byte count (usually 60) to initiate the interchange.

In response to a send or receive command from the host computer, the CU transfers the message text in a sequence of 60-byte blocks and a partial block to carry the last segment. A CP shared memory has a capacity of 128 bytes (two blocks) for output data and the same for input data. CPs use reserved ICWs to notify the CU when an output block has been transmitted to the terminal or when an input block has been filled with data received from the terminal. The control unit performs the multiplexing function to interleave the block transfer with the CPs in the PDCC.

As noted in the previous section, one group of I/O command words is reserved for block transfer operations. The block transfer group (H1 = F) is used by the CU to notify a port that a block of output data has been loaded into its stored memory or that a block of input data has been removed. A CP uses this group to signal the CU that the transmission of an output block is complete or that the reception of an input block is complete and the block is available for transfer from the shared memory.

Table 4-3 describes the OCWs in the block transfer group, and Table 4-4 describes the ICWs.



TABLE 4-3. BLOCK TRANSFER GROUP FOR OUTPUT COMMAND WORDS (H1 = F)

H2	MNEMONIC	FUNCTION	USE OF MODIFIER FIELD
0	MSG-OUT	Notify CP that an output data block has been loaded into the shared memory and that the block consists of a full message	Number of bytes in the block
1	BLK-OUT/ SOM	Notify CP that an output data block (60 bytes) has been loaded into the shared memory and that the block is the first in a message text	M = 3C (hex)
2	BLK-OUT	Notify CP that a 60-byte output data block has been loaded into the shared memory (does not start or end a message)	M = 3C (hex)
3	BLK-OUT/ EOM	Notify CP that an output data block has been loaded into shared memory and that it is the last data block in the message	Number of bytes in the message
4	BLK-IN	Notify CP that an input data block has been removed from the shared memory	Number of bytes in the message
5 ↓ F	Unassigned		

TABLE 4-4. BLOCK TRANSFER GROUP FOR INPUT COMMAND WORDS (H1 = F)

H2	MNEMONIC	FUNCTION	USE OF MODIFIER FIELD
0	MSG-IN	Notify CU that an input data block is ready for transfer from the shared memory and that the block consists of a full message	Number of bytes in the block
1	BLK-IN/ SOM	Notify CU that a 60-byte input data block is ready for transfer from the shared memory and that the block is the first in a message	M = 3C (hex)
2	BLK-IN	Notify CU that a 60-byte input data block is ready for transfer from the shared memory (the block does not start or end a message)	M = 3C (hex)
3	BLK-IN (EOM)	Notify CU that an input data block is ready for transfer from the shared memory and that the block is the last in the message	Number of bytes in the message
4	BLK-OUT	Notify CU that an output data block has been removed from the shared memory and transmitted to the terminal	Number of bytes in the message
5 + F	Unassigned		

#### 4.3 FUNCTIONAL DESCRIPTION OF A COMMUNICATION PORT

The operation of a communication port is governed by the stored program of the CP microcomputer. This program uses inputs from the control unit (supplied via the OCR) to determine the mode of operation, set system variables, and select the procedures for execution. In general, the CP communicates with a terminal by sending and receiving messages in response to write (send) and read (receive) instructions issued by an application program and passed along by the PDCC driver and the control unit. Depending on the features of its stored program, a communication port may also interchange prescribed control messages with the terminal in response to special output commands originating from the application program.

Figure 4-3 shows the elements of a CP that relate most directly to communication functions. The purpose of the shared memory is to buffer the text of messages being sent/received by the CP microcomputer. The communications interface performs the conversion from the byte serial orientation of the microcomputer I/O to the bit-serial feature of the communication lines to the terminal. Under program control, it also generates and detects control signals for interchange of data with the data communication equipment that transmits signals between the terminal site and the host computer site. The OCR and ICR, as discussed previously, are the means for the CU and the CP to interchange status and control information.

Conforming to the line and terminal protocols imbedded in its stored program, the CP microcomputer constructs the communication messages destined for the terminal. It reads data from the output channel of the stored memory to use as the message text. The microcomputer effects message transmission by successively writing data and control bytes to the communications interface at a rate determined by the communication equipment's signaling speed. Similarly, it receives communication messages by successively reading data and control bytes from the communications interface as they arrive. The microcomputer also extracts the text from the message and successively writes it into the input channel of the shared memory one byte at a time.

## COMMUNICATION PORT

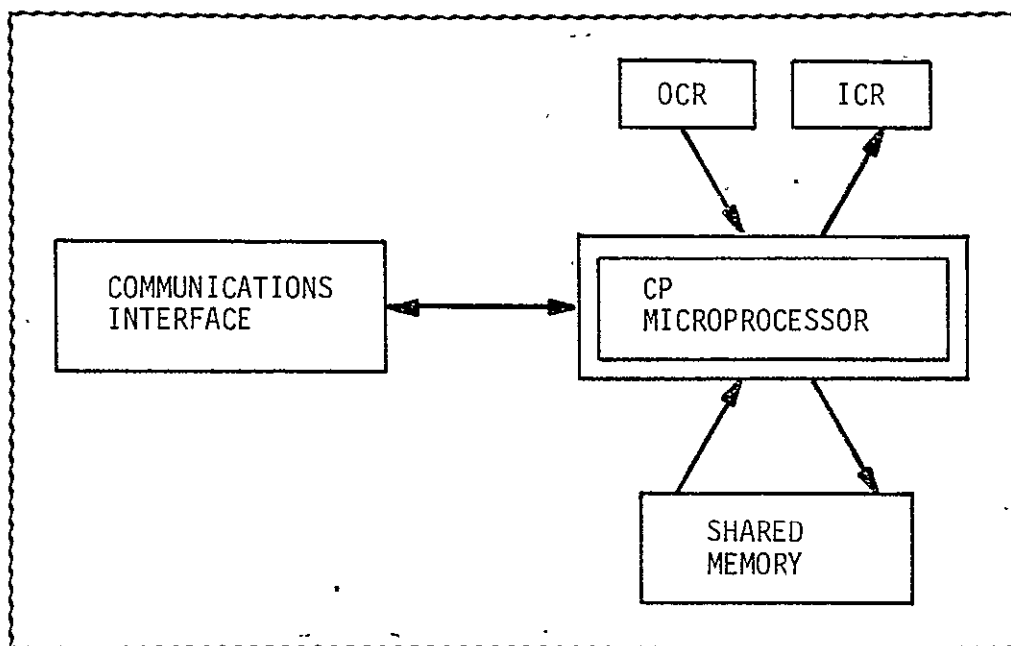


FIGURE 4-3. DATA FLOW INTERNAL TO COMMUNICATION PORT

The CP stored program also contains the code for the bookkeeping functions related to the transfer of data blocks between the CP shared memory and the host computer's memory. In its RAM, the microprocessor maintains a table with entries for each data block in the input and output channels of the shared memory. Each entry includes the byte count for the block and a flag to indicate whether the block is the first, intermediate, or last in a message text. Entries are added/deleted as OCWs from the block transfer group appear in the OCR to notify the CP that data blocks have been placed in the output channel or removed from the input channel of the shared memory. Using the ICR, the CP microcomputer notifies the CU when it has removed all data from the oldest block in the output channel or has filled the newest block in the input channel.

## 5. RECOMMENDED INVESTIGATIONS

The requirements for the PDCC, as presented in this document, were motivated by the following considerations, among others:

- A system can be implemented with a smaller number of stations than the maximum of 23.
- The number of wires in the Databus is as few as practical.
- A plug-in unit has no external connections other than provided by the contacts on the Databus connector.
- The Databus address scheme makes provision for as many as four communication ports at a station.
- The mechanical layout of the crate is based on a standard configuration.

These attributes have apparent benefits. There are alternatives that merit further investigation, however.

The data lines in the Databus are bidirectional, consistent with minimizing the number of wires in this structure. We recommend that a study investigate the alternative approach of multiple unidirectional lines to assure that no significant benefits are being sacrificed. Further, the address assignments for registers on the Databus and the gating logic specified for each station are compatible with the Intel 8012 I/O port that was used to specify the requirement. We also recommend consideration of other I/O chips to confirm that they also are compatible with the addressing and decoding scheme:

The pin assignments for the interface lines were not specified in the requirements for the Databus connector. They should be defined in terms of the RS 232 connector for four situations: one CP at the station, two CPs at the station, three CPs at the station, and four CPs at the station.

# NTS

from NATIONAL TECHNICAL INFORMATION SERVICE

**Product Liability: Industry Study. Volumes I and II**  
PB-265 542/PSK 886 p PC\$21.25/MF\$3.00

**An Atlas of Radiation Histopathology**  
TID-26676/PSK 234 p PC\$7.60/MF\$3.00

**The Medical School Admissions Process. A Review  
of the Literature, 1955-76**  
PB-263 962/PSK 193 p PC\$7.50/MF\$3.00

**Integrated Energy Vocabulary**  
PB-259 000/PSK 459 p PC\$22.50/MF\$3.00

**Nuclear Air Cleaning Handbook. Design, Construction, and Testing of High-Efficiency Air Cleaning Systems for Nuclear Application**  
ERDA-76/21/PSK 300 p PC\$9.75/MF\$3.00

**Evaluation of the Air-To-Air Heat Pump for Residential Space Conditioning**  
PB-255 652/PSK 293 p PC\$9.25/MF\$3.00

**Federal Information Processing Standards Register:  
Guidelines for Documentation of Computer  
Programs and Automated Data Systems. Category:  
Software. Subcategory: Documentation**  
FIPS-PUB38/PSK 55 p PC\$4.50/MF\$3.00

**Life Cycle Costing Emphasizing Energy Conservation: Guidelines for Investment Analysis**  
ERDA-76/130/PSK 117 p PC\$5.50/MF\$3.00

**Analysis of Projected Vs. Actual Costs for Nuclear and Coal-Fired Power Plants**  
FE-2453-2/PSK 38 p PC\$4.00/MF\$3.00

**Quality Criteria for Water**  
PB-263 943/PSK 537 p PC\$13.00/MF\$3.00

**Passive Solar Heating and Cooling Conference and  
Workshop Proceedings May 18-19, 1976,  
University of New Mexico, Albuquerque, New  
Mexico**  
LA-6637-C/PSK 352 p PC\$10.50/MF\$3.00.

**Predicting the Performance of Solar Energy Systems**  
AD-A035 608/PSK 45 p PC\$4.00/MF\$3.00

**NIOSH Analysis Methods for Set J**  
PB-263 959/PSK 128 p PC\$6.00/MF\$3.00

**Flat-Plate Solar Collector Handbook: A Survey of Principles, Technical Data and Evaluation Results**  
UCID-17086/PSK 96 p PC\$5.00/MF\$3.00

## How to Order

When you indicate this method of payment, please note if a purchase order is not accompanied by payment, you will be billed an additional \$5.00 *ship and bill* charge. And please include the card expiration date when using American Express.

Normal delivery time takes three to five weeks. It is vital that you order by number or your order will be manually filled, insuring a delay. You can opt for *priority mail* for \$3.00 outside North American continent charge per item. Just check the *priority mail* box. If you're really pressed for time, call the NTIS Rush Handling Service (703) 557-4700. For a \$10.00 charge per item, your order will be sent *priority mail* within 48 hours. Or, you can pick up your order in the Washington

Information Center & Bookstore or at our Springfield Operations Center within 24 hours for a \$6.00 per item charge.

You may also place your order by telephone or if you have an NTIS Deposit Account or an American Express card order through TELEX. The order desk number is (703) 557-4650 and the TELEX number is 89-9405.

Thank you for your interest in NTIS. We appreciate your order.

## METHOD OF PAYMENT

- ☐ Charge my NTIS deposit account no. \_\_\_\_\_
- ☐ Purchase order no. \_\_\_\_\_
- ☐ Check enclosed for \$ \_\_\_\_\_
- ☐ Bill me. Add \$5.00 per order and sign below. (Not available outside North American continent.)
- ☐ Charge to my American Express Card account number \_\_\_\_\_

[illegible]

Card expiration date \_\_\_\_\_

Signature\_\_\_\_\_

- ☐
- Priority mail requested

**Clip and mail to:**

**NTTS**

**National Technical Information Service  
U.S. DEPARTMENT OF COMMERCE  
Springfield, Va. 22161  
(703) 557-4850 TELEX 89-9405**

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY, STATE, ZIP \_\_\_\_\_

Item Number	Quantity		Unit price*	Total Price*
	Paper Copy (PC)	Microfiche (MF)		

All prices subject to change. The prices above are accurate as of 3/31/78  
Foreign Prices on Request.

Sub Total  
Additional Charge  
Enter Grand Total